

P. ENT COOPERATION TREA.

PCT

NOTIFICATION OF ELECTION

(PCT Rule 61.2)

From the INTERNATIONAL BUREAU

To:

Commissioner
 US Department of Commerce
 United States Patent and Trademark
 Office, PCT
 2011 South Clark Place Room
 CP2/5C24
 Arlington, VA 22202
 ETATS-UNIS D'AMERIQUE
 in its capacity as elected Office

Date of mailing (day/month/year) 16 November 2000 (16.11.00)	
International application No. PCT/GB00/01135	Applicant's or agent's file reference A25771 WO
International filing date (day/month/year) 24 March 2000 (24.03.00)	Priority date (day/month/year) 31 March 1999 (31.03.99)
Applicant POWERS, Simon, Julian et al	

1. The designated Office is hereby notified of its election made:

☒ in the demand filed with the International Preliminary Examining Authority on:
 20 October 2000 (20.10.00)

☐ in a notice effecting later election filed with the International Bureau on:

2. The election ☒ was

☐ was not

made before the expiration of 19 months from the priority date or, where Rule 32 applies, within the time limit under Rule 32.2(b).

The International Bureau of WIPO 34, chemin des Colombettes 1211 Geneva 20, Switzerland	Authorized officer S. Mafla
Facsimile No.: (41-22) 740.14.35	Telephone No.: (41-22) 338.83.38



INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁷ :

H04L 29/06

A1

(11) International Publication Number:

WO 00/62506

(43) International Publication Date:

19 October 2000 (19.10.00)

(21) International Application Number: PCT/GB00/01135

(22) International Filing Date: 24 March 2000 (24.03.00)

(30) Priority Data:

9907445.2

31 March 1999 (31.03.99)

GB

99305556.5

13 July 1999 (13.07.99)

EP

(71) Applicant (for all designated States except US): BRITISH TELECOMMUNICATIONS PUBLIC LIMITED COMPANY [GB/GB]; 81 Newgate Street, London EC1A 7AJ (GB).

(72) Inventors; and

(75) Inventors/Applicants (for US only): POWERS, Simon, Julian [GB/GB]; 2A Redan Street, Ipswich, Suffolk IP1 3PQ (GB). HINDS, Michael, Reuden [GB/GB]; 25 Spriteshall Lane, Trimley St Mary, Felixstowe IP11 9QY (GB).

(74) Agent: GARRISON, Christopher, Sinclair; BT Group Legal Services, Intellectual Property Dept., 8th floor, Holborn Centre, 120 Holborn, London EC1N 2TE (GB).

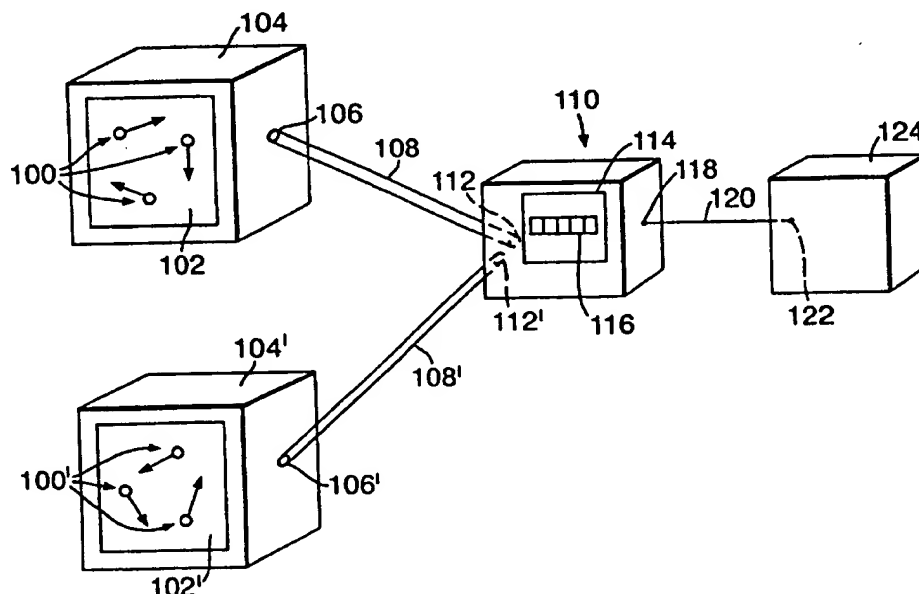
(81) Designated States: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published

With international search report.

Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: PACKET MESSAGING METHOD AND APPARATUS



(57) Abstract

According to the invention, each packet message in a sequence of packet messages is sent with an associated packet message priority label in accordance with a predetermined cyclic sequence of such labels. These priority labels each represent one of a plurality of priority levels and the position of each label in the cyclic sequence is such that the number of consecutive lower priority labels between that label and the nearest label of equal or higher priority is substantially maximised. In this way, if packet messages are dropped from the sequence due to congestion on a priority basis, the packet messaging method and apparatus according to the invention ensure that, as far as possible, regular update packet messages are still received from all packet message sources.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakhstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

PACKET MESSAGING METHOD AND APPARATUS

The present invention relates to a method of and apparatus for the management of packet messaging. The invention has application, in particular, in the optimal management of a certain class of packet messaging over a low bandwidth link.

A wide variety of situations are known where a source entity transmits messages over a communications link to a receiving entity. A well known subset of this variety of situations involves the communications link over which the source entity transmits its messages having an insufficient bandwidth to be able to carry the volume of messages intended for the receiving entity.

One crude solution to this problem would simply be to send as many messages as can be sent at any given time and then to discard any remaining 'overflow' messages. Whilst having the merit of simplicity, this solution cannot provide an efficient utilisation of the communication link. A sudden burst of activity may lead to a flood of messages with the overflow messages being discarded. Following the burst of activity however, the communications link may return to being only lightly loaded.

An improved solution would provide for the buffering of the overflow messages. The overflow messages would not then simply be discarded if they could not be sent at any given time, but could be buffered and sent following the burst of activity, when the communications link had returned to being only lightly loaded. It will be clear that there will be limits to the size of the message buffer, however, such that it might be imagined that the buffer itself might overflow. It will also be clear that there is no substantive discrimination between the messages, all messages being treated as being of equal importance.

A different approach altogether may be applied in certain circumstances where the messages may be considered as being of different importance. As and when congestion on the communications link between the source entity and the receiving entity occurs, more important messages may be transmitted over the communications

link whereas less important messages may be discarded, or buffered as discussed above.

5 A well known example of such an approach with a single source entity is the concept of 'layered video coding'. Such an approach could just as well be applied to a number of competing source entities however, where the source entities and hence their respective messages are ranked according to their importance.

10 A problem having somewhat different characteristics from the preceding situation is, rather, one where there are a number of competing yet equally important source entities all transmitting the same type of message. An example might be a number of source entities that transmit messages as to some aspect of their state. Mobile entities might, for example, transmit messages as to their location. Likewise, sensor entities might transmit messages as to ambient temperature or pressure.

15

A more familiar example might be that of a networked computer game such as 'Quake' (created by ID Software of Mesquite, TX, USA), where mobile creatures are created in a three-dimensional virtual environment running on one or more server computers. The positions of these creature source entities may be messaged to game
20 entities hosted on client computers connected to the server or servers as appropriate.

Typically, what will be most important for the entity receiving such periodic state update messages, is to ensure that regular updates are received from all the source entities. If a number of source entities are sending messages with state updates, it
25 will typically be more important to receive regular state updates from all the source entities than to receive all the state updates from a subset of the source entities and few, if any, state updates from the remaining source entities. This is to say that it is typically more important to keep down the maximum possible state error of source entities rather than the average state error of all the source entities.

30

To use the example of a game such as Quake again, if a game player's character is surrounded by highly mobile hostile creatures, it may well be more important for the game player's character to know where all the relevant creatures are (or rather where

they all were a short time ago) to within a relatively small error rather than knowing where a subset of the creatures are to within a very small error (where they are now) with the remainder of the (highly mobile hostile) creature's positions only being known with a relatively large error (where they were a while ago).

5

It will be clear that a simple hierarchically oriented scheme such as a layered video coding scheme cannot provide for such a situation. As indicated above, in such a scheme all available information may be required from the most important source entities at the expense of receiving little, if any, information from the less important
10 source entities whereas having regard to the problem under consideration, at least some information is required from all the source entities, each being nominally of the same importance.

Thus, according to one aspect of the present invention, there is provided a packet
15 message source comprising: means arranged to include a respective packet message payload in each packet message of a sequence of packet messages; means arranged to associate a priority label with each successive packet message in said sequence in accordance with a predetermined cyclic sequence of such labels; said priority labels each representing one of a plurality of priority levels and the position of each label in
20 the cyclic sequence being such that the number of consecutive lower priority labels between that label and the nearest label of equal or higher priority is substantially maximised; and means arranged to send such packet messages.

Advantageously, this will ensure that, when a number of such packet message
25 sources are sending such packet messages and must contend for limited bandwidth such that packets are dropped as necessary in accordance with their priority labels, as far as possible packet messages representing regular updates from all the sources can be sent to a receiving client, rather than all the updates from a subset of the sources and few, if any, updates from the remaining sources as with the prior art.

30

A packet message system, a method of packet messaging and a method of operating a packet messaging system are also provided.

Several embodiments of the invention will now be described, by way of example, having regard to the accompanying drawings, in which:

Figure 1 illustrates a first embodiment according to the invention;

5

Figure 2 illustrates a general purpose computer;

Figure 3 illustrates a packet message structure;

10 Figures 4A and 4B illustrates first procedural flowcharts;

Figure 5 illustrates a second procedural flowchart, having regard to a second embodiment according to the invention;

15 Figure 6 illustrates a third procedural flowchart, having regard to the second embodiment according to the invention;

Figure 7 illustrates a first graph, showing performance results for the second embodiment;

20

Figure 8 illustrates a second graph, showing further performance results for the second embodiment;

Figures 9A and 9B illustrate a fourth procedural flowchart, having regard to a third
25 embodiment according to the invention;

Figure 10 illustrates a third graph, showing performance results for the third embodiment;

30 Figure 11 illustrates a fourth graph, showing further performance results for the third embodiment; and

Figure 12 illustrates a fourth embodiment according to the invention.

A first embodiment according to the invention will now be described having regard to Figures 1, 2, 3 and 4. Figure 1 illustrates in schematic form, a first embodiment architecture according to the invention. Figure 4 illustrates a procedural flowchart of the functioning of this first embodiment.

Each one of a first group of source entities, each source entity having an associated state, creates and sends packet messages containing information as to its current state. The destination entity or entities to which the messages are sent will be discussed below. The source entities can take a wide variety of forms; one example of such a form is that of an entity hosted within a software process on a computer.

Figure 1 thus illustrates, by way of example, a group of source entities 100 as being hosted within a first software process 102 on a first server computer 104.

Such a computer 104 is as illustrated in Figure 2 and will typically have at least a central processing unit (CPU) 202, read-only memory (ROM) 204, random-access memory (RAM) 206, a storage device such as a hard disk 208, a device for reading from and writing to storage media such as a floppy disk drive 210 for reading from and writing to floppy disks and input and output ports 212 for connection to other devices or communications networks.

The computer 104 may utilise any suitable operating system, well known examples being Microsoft Windows™ NT, Linux or any one of the other many versions of Unix. Application programs may be written in any of many well known suitable languages in which to write application programs, one well known example of which is C++. Such an operating system and application programs may be loaded onto the storage device 208 of the computer 104.

Aspects of the functionality disclosed in accordance with the embodiments of the invention may be implemented as software application programs to be executed by the computer 104. This software application program may then be stored in any suitable computer readable storage media form, for example on a floppy disk 214, for

loading into the computer 104, via the floppy disk drive 210, for execution. A well known alternative would be to store the software application on a CD-ROM (not shown) for loading into the computer 104 via a CD-ROM drive (not shown) for execution. A further well known alternative would be to download the software application program over a suitable network (not shown), for execution by the computer 104.

In this embodiment the computer 104 has one or more software application programs loaded onto it which, when executed, will cause the computer 104 to operate as such a host for source entities.

Often, there will be more than one such server computer. Accordingly a second group of source entities 100' is illustrated, hosted in a second software process 102' on a second server computer 104'.

15

As with the example of Quake above, the source entities could, for example, be creatures inhabiting a three-dimensional virtual environment, with different portions of the virtual environment running as processes on one or more of the server computers. By way of a simple example, a group of creatures in one room of a virtual environment might be represented by the source entities running in a process on one server computer and another group of creatures in another room of the virtual environment might be represented by the source entities running in another process on a second server computer.

As indicated above, each source entity 100 has an associated state; in this example each character will have at least an associated position within the virtual environment. Thus, each character, moving about in the three-dimensional virtual environment, may create and send packet messages as to their respective current positions in the three-dimensional virtual environment.

30

An example of the structure of such a packet message is illustrated in Figure 3. As is conventional, the packet message 300 has both a header portion 302 and a payload

portion 304. It is to be noted that not all the fields in the packet header 302 will be simultaneously required for all the embodiments of the invention.

A first field 306 in the packet header 302 contains an indication of the identity of the source entity. A second field 308 in the packet header 302 contains an indication of the identity of the intended destination entity. With only one destination in this first embodiment, this field need not be set. A third field 310 in the packet header 302 contains an indication of assigned priority, on a given priority scale.

10 According to the invention, the third field 310, identifying a priority setting, will be set as follows. Each individual packet message is assigned a priority label, each priority label representing one of a plurality of priority levels. Successive packets, considered in the sequence in which they were generated by each packet message source, are allocated one of these priority labels in accordance with a predetermined
15 cyclic label sequence. Crucially, the position of each label in this cyclic sequence is such that the number of consecutive lower priority labels between it and the nearest label of equal or higher priority is substantially maximised.

The cyclic priorities thus vary such that if packets were discarded on a priority basis,
20 the remaining packets in the notional sequence would be left as evenly spaced as possible. It will be noted that sequence lengths of powers of two are advantageous in this respect.

For a simple example with a priority scale of 0 (lowest priority) to n (highest priority)
25 and where $n=7$, a corresponding suitable cyclic variation in priority is a sequence of eight messages (equally spaced in time) with priorities set at 0, 7, 2, 4, 1, 6, 3 and 5 respectively. The 0, 7, 2, 4, 1, 6, 3, 5 sequence will ensure that as successive low priorities have to be dropped the remaining packets will be as evenly spaced as possible. By way of example, if half the packets had to be dropped from such a
30 sequence of eight messages due to a traffic load of twice link capacity (thus becoming [], 7, 2, 4, 1, 6, 3, 5 then [], 7, 2, 4, [], 6, 3, 5 then [], 7, [], 4, [], 6, 3, 5, and finally [], 7, [], 4, [], 6, [], 5) it will be seen that the remaining packets are

still equally spaced in time and will neatly provide updates every two temporal periods, rather than every temporal period as with the original sequence.

5 A fourth field 312 in the packet header 302 contains an indication of a so-called time-to-live. The time-to-live would be indicated as a given period of time, ranging, for example, up to a couple of hundred milliseconds, rather than being indicated as a given number of link hops is often the case in packet networks. In the first embodiment this field is not set.

10 A fifth field 314 in the packet header 302 indicates a payload type. The payload type might be, for example, that for a position indication. In the first embodiment this field is not set.

The packet payload portion 304 carries data. In this example the typical payload will
15 be a position indication.

Having regard to Figure 4A, in a first step 400 then, a source entity creates and sends a stream of such packet messages, carrying positional updates in the packet payloads and cyclically varying priorities in the fourth field of the packet headers.

20

Returning to Figure 1, the first and second server computers 104, 104' will have first and second server computer output ports 106, 106' respectively. These respective first and second server computer output ports 106, 106' are connected by respective first and second high-bandwidth links 108, 108' to a communications link interface,
25 link management computer 110. Such high-bandwidth links 108, 108' may typically take the form of, for example, Fast Ethernet, with an associated bandwidth of, for example, 100Mbits per second.

The link management computer 110 may take the form of a general purpose
30 computer as illustrated in Figure 2. The link management computer 110 may utilise any suitable operating system, well known examples being Microsoft Windows™ NT , Linux or any one of the other many versions of Unix. Application programs may be written in any of many well known suitable languages in which to write application

programs, one well known example of which is C + + . Such an operating system and application programs may be loaded onto a storage device 208 of the computer 110.

The functionality disclosed in accordance with the embodiments of the invention may
5 be implemented as software application programs to be executed by the link management computer 110. This software application program may then be stored in any suitable computer readable storage media form, for example on a floppy disk 214, for loading into the computer 110, via the floppy disk drive 210, for execution. A well known alternative would be to store the software application on a CD-ROM
10 (not shown) for loading into the computer 110 via a CD-ROM drive (not shown) for execution. A further well known alternative would be to download the software application program over a suitable network (not shown), for execution by the computer 110.

15 The link management computer 110 has at least one input port 112, through which the link management computer receives the packet messages sent by the source entities. In this example, with two server computers 104, 104' , first and second link management computer input ports 112, 112' are illustrated. it will be appreciated that these first and second link management computer input ports 112, 112' may be
20 either separate physical ports or a shared physical port separated at the software level.

In a second step 402, the packet messages sent by the source entities hosted in the first and second server computers 104, 104' are thus sent to the link management
25 computer 110 via the respective first and second server computer output ports 106, 106' , the first and second fast links 108, 108' and the first and second link management computer input ports 112, 112' . The links between the respective server computers and link management computer are of a sufficiently high bandwidth that the packet messages sent by the source entities can be carried to the link
30 management computer with a delay which is insignificant compared to the further downstream delays.

In this embodiment the link management computer 110 has one or more software application programs loaded onto it which, when executed, will cause the link management computer to run a Link Manager process 114. The appropriate steps of the procedural flowcharts will be carried out through execution of the software application programs.

In a third step 404, the Link Manager process 114 receives the packet messages arriving at the first and second link management computer input ports 112, 112'. In a fourth step 406, the Link Manager process 114 reads the priority setting from the fourth field 312 of the packet headers 302 of the received packets.

In a fifth step 408, the Link Manager process 114 sorts the packet messages that arrived at the first and second link management computer input ports 112, 112' into a simple sorted linear packet queue 116 ordered on priority. The packet message might, for example, be sorted into the queue 116 at the rear of the group of packet messages with the same priority setting. It will be appreciated that, by way of an alternative, the portions of the simple queue 116 with the same priority settings could just as easily be queued, in an optimised implementation as separate queues for each priority setting.

It will be appreciated that in a further step (not shown), the length of the packet queue 116 could be checked, and packet messages being stored as to increase the length of the queue 116 beyond a predetermined limit queue length could be discarded.

The link management computer 110 also has at least one output port 118.

The output port 118 of the link management computer 110 is connected, by means of a link 120 of known Quality-of-Service and, in particular, a known low-bandwidth, to, for example, an input port 122 of a receiving entity host computer 124. When the Link Manager process 114 sends the packet message at the head of the priority queue 116 to the link management computer output port 118, the message is then

transmitted over the communication link 120 to the receiving entity host computer 124, which can then pass the message to the receiving entity (not shown).

Having regard to Figure 4B, in a step 410, the Link Manager will test to see if there is spare capacity on the low bandwidth link to send the packet message at the head of the queue 116. If there is, in a further step 412, the Link Manager process 114 will send the packet message at the head of the priority queue 116 to the link management computer output port 118 and out onto the link 120. If there is not, then step 410 will be repeated. This is done as an intelligent selection of packet messages has effectively been made at the link manager according to the invention through the cyclic priority scheme; overloading the link might simply cause a return of random packet message loss, losing the fruits of the invention.

It will be appreciated that the sending of the packet message at the head of the queue 116 will thus involve a loop process. The Link Manager process 114 must wait until the current head of queue packet message has been sent before attempting to send the next head of queue packet message. This might be achieved, for example, with the Link Manager process ascertaining the size of the packet message at the head of the queue 116, which can then be used, with a knowledge of the Bandwidth of the link 120, to compute how long the head of queue packet message will take to be sent. By way of an alternative, the packet messages for sending could be sent from a buffer, into which a following packet message cannot be written till the previous packet message has cleared the buffer.

A typical low-bandwidth link 120 might, for example, be a 33.6 kbits per second Modem link over a Public Switched Telephone Network (PSTN) connection or a 128 kbits per second Integrated Services Digital Network (ISDN) connection.

By way of example, the receiving entity host computer 124 might run a process which hosts an entity intended to receive the messages sent by the source entities. It will be appreciated however, that instead of the receiving entity host computer, a receiving entity might instead take a wide variety of forms, not even having to take a computationally capable form, for example, a simple data store.

For as long as the total bandwidth of the packet message traffic received by the link management computer 110 at any given time has been less than the known low-bandwidth of the communication link 120, all packet messages placed in the Link

5 Manager process queue 116 will be sent out on the link 120 with no queuing delay.

The receiving entity will receive all the packet messages sent from all the source entities with the lowest possible delay (which will result only from the network delay).

10 It may be the case, however, that increasing numbers of source entities 100, 100' become active, resulting in a commensurately increased flow of messages.

Alternatively, the source entities 100, 100' themselves may become more active, resulting in the need for state update messages with reduced periodicity. In either case, as the amount of packet message traffic increases, there will come a point

15 when the total bandwidth of the incoming packet message traffic exceeds the bandwidth of the communication link 120. At this point it will no longer be possible to send on simultaneously all the packet messages intended for the receiving entity.

The packet messages will now begin to be queued by the Link Manager process 114,

20 ordered, as indicated above, on the basis of priority. As new packet messages arrive at the link management computer 110 they will be sorted into the section of the queue appropriate to the priority setting carried in their packet header 302.

The discussion as to the cyclic priority settings of the sequences of packet messages

25 sent out from the source entities 100, 100' will be recalled. The example was provided above, of the management of the selecting of packets for sending on a link

with a traffic load running at twice link capacity. If half the packets were dropped from a sequence of eight messages with cycled priority sequences of the form 0, 7, 2, 4, 1, 6, 3, 5 due to a traffic load of twice link capacity (thus becoming [], 7, 2, 4,

30 1, 6, 3, 5 then [], 7, 2, 4, [], 6, 3, 5 then [], 7, [], 4, [], 6, 3, 5, and finally [], 7, [], 4, [], 6, [], 5) it was seen that the remaining packets were still equally spaced in time and would neatly provide updates every two temporal periods, rather than every temporal period as with the original sequence.

Thus when the Link Manager process 114 begins to queue packet messages on a priority basis, it will effectively begin to drop (or rather queue) the lower priority messages whilst sending on the higher priority messages, in an cyclic order similar to the example given above. In this way, a link management computer 110 according to this embodiment of the invention, with an increasingly congested link 120 , will be able to continue to send regular state updates from all the source entities 100, 100'. Naturally, these updates, whilst remaining regular, will become less and less frequent with rising low-bandwidth link 120 congestion, but this gradual falling away of frequency indicates the graceful manner in which the degradation of service takes place.

Simulated performance results for this first embodiment will be discussed below having regard to a comparison with the performance results of the third embodiment illustrated in Figures 10 and 11.

A second embodiment according to the invention will now be discussed having regard to Figures 1, 2, 3, 5, 6, 7 and 8. This second embodiment differs from the first only in terms of the structure of the packet messages 300 sent by the source entities 100, 100' and in the functionality of the Link Manager process 114, so reference should again be made to Figures 1, 2 and 3 where appropriate. Figures 5 and 6 illustrate procedural flowcharts of aspects of functioning of this second embodiment according to the invention. Again the appropriate steps of the procedural flowcharts will be carried out through the execution of the software application programs running on the link management computer 110.

According to this second embodiment and having particular regard to Figures 1 and 2, the source entities 100, 100' transmit packet messages 300 with the following structure. The first field 306 in the packet header 302 will be set with the identity of the sending source entity. The second field 308 in the packet header 302, identifying a destination entity, need not be set as there exists only one destination entity in this example.

The third field 310, identifying a priority setting, will be set as with the first embodiment. Each individual packet message is assigned a priority setting on a scale of for example, 0 (lowest priority) to n (highest priority). Again crucially, a sequence of $n + 1$ such messages is sent out from the source entity, not with a fixed priority assigned to the source entity in question, but with cyclically varying priorities. The cyclic priorities vary such that if packets were discarded on a priority basis, the remaining packets in the sequence would be left as evenly spaced as possible.

Again, for a simple example where $n = 7$, a corresponding suitable cyclic variation in priority is a sequence of eight messages with priorities set at 0, 7, 2, 4, 1, 6, 3 and 5 respectively. As indicated above, the 0, 7, 2, 4, 1, 6, 3, 5 sequence will ensure that as successive low priority packet messages have to be dropped, the remaining packet messages will be as evenly spaced as possible.

In this second embodiment the fourth field 312 will be set with a time-to-live period. It will be appreciated that rather than all source entities needing to have the same packet message sending periodicity, varying time-to-live indications allow each source entity to set their own periodicity. For example, as a result of its nature, a given source entity might only need to send updates every 20 nominal time periods, in which case a time-to-live period could be set at 20 nominal time periods. After 20 such nominal time periods, the given source entity will send out a new update so the older update may be discarded.

In this second embodiment, the fifth field 314 in the packet header 302, giving an indication of the packet payload type, is not set.

Streams of such packet messages, bearing cyclic priority settings, as discussed above, are sent out from the source entities 100, 100'. These packet messages are then passed, via the respective first and second server computer output ports 106, 106' and first and second links 108, 108', to the link management computer 110.

Having regard to Figure 5, in a first step 500, after these packet messages are received at the first and second link management computer input ports 112, 112', they are passed to the Link Manager process 114.

- 5 The Link Manager process 114 will again maintain a simple packet message queue 116, ordered on priority, as with the first embodiment.

The Link Manager process 114 is provided with a clock (not shown) keeping a local time measured cyclically in, for example milliseconds. In a second step 502, the Link
10 Manager process 114 reads the time-to-live setting in the fourth field 312 of the respective packet headers 302 and in a third step 504, adds the local clock time to the time-to-live setting assigned by the source entity, to create an adjusted time-to-live. In a fourth step 506, this adjusted time-to-live is then written back into the fourth field 312 of the packet header 302.

15

It will be appreciated that, by way of an alternative, the packet message and an associated arrival time could instead be stored together in the queue 116. This would have the advantage of not writing over the source entity set time-to-live but would have the disadvantage of a commensurately larger queue memory requirement.

20

It will be appreciated that the only significant delay which may be experienced by a given packet message will be that resulting from being held in a queue. If, by way of the example used above, a given packet message is received by the Link Manager process 114 with a time-to-live indication of 20 nominal units and the Link Manager
25 process local time was $t = 100$ nominal units, then the adjusted time-to-live written into the fourth field of the packet header would be 120 nominal units. If, for reasons discussed below, this packet message was queued for more than the 20 nominal units lifetime, by which time the Link Manager process local time would be $t = 120$ nominal units, then the packet message should be timed-out. The source entity
30 should have sent, and the Link Manager process 116 should have received, another packet message by this time.

In a fifth step 506, the Link manager process 114 also reads each priority setting in the third field 310 of the respective packet headers 302 and then, in a sixth step 510, sorts the packet messages into the simple queue 116 ordered on the basis of the priority setting. Again, the packet message might, for example, be sorted into the queue 116 at the rear of the group of packet messages with the same priority setting. As indicated above, again, by way of an alternative, the portions of the simple queue 116 with the same priority settings could be queued separately.

The process illustrated in Figure 6 is to be understood as being executed in parallel with that illustrated in Figure 5.

Having regard to Figure 6, in a first step 600, as each new packet message moves up to the head of the queue 116 for sending onto the link 120, the Link Manager process 114 will again read the adjusted time-to-live setting from the fourth field 312 of the packet header 302 of the packet message at the head of the queue 116.

In a second step 602, the Link Manager 114 will compare this adjusted time-to-live with the Link Manager process local time. In a third step 604, if the Link Manager process local time has passed the adjusted time-to-live of the packet message, then that packet message must be considered as being timed-out and will be discarded by the Link Manager process 114. If, however, the Link Manager process local time has not passed the adjusted time-to-live of the packet message, then the packet message will not have been queued for longer than its assigned time-to-live and in a fourth step 606, will next be sent, via the output port of the link management computer 118, onto the link and to the receiving entity at the client 120. The Link Manager process 114 will again test the link 120 for capacity to send the packet message at the head of the priority queue 116 in doing so, as with the first embodiment.

It will be appreciated that for optimal utilisation of the bandwidth available on the low-bandwidth link, a step of header compression (not shown) may take place. Portions of the header 302 no longer necessary since the packet message is to be sent out over the low-bandwidth link 120, for example, the priority setting, may be stripped off.

As with the first embodiment, for as long as the instantaneous total bandwidth of the packet message traffic received by the link management computer 110 has been less than the known (low) bandwidth of the communication link 120, all packet messages placed in the Link Manager process queue 116 will be sent out on the link with no queuing delay. The receiving entity 124 will receive all the packet messages sent from all the source entities 100, 100'. Again, however, as with the first embodiment, as the amount of packet message traffic increases, there will come a point when the total bandwidth of the incoming packet message traffic exceeds the bandwidth of the communication link 120. At this point it will no longer be possible to send on simultaneously all the packet messages intended for the receiving entity.

The second embodiment according to the invention will provide all the advantages of the first embodiment according to the invention in terms of the cyclic priority settings being able to deliver an optimal and graceful degradation of service as the total traffic bandwidth exceeds the low-bandwidth link capacity. The second embodiment according to the invention will, however, provide yet further advantages in terms of performance over the first embodiment, as will now be explained.

The assignment of a time-to-live for each message allows the avoidance of an erroneous re-ordering of messages as follows.

A first positional packet message is sent from a given source entity at a nominal $t=0$ with a priority setting of 0. A second packet message is sent at nominal $t=1$ with a priority setting of 7. A third packet message is sent at nominal $t=2$ with a priority setting of 2. If, due to a degree of congestion on the low bandwidth link 120, the first packet message, with the lowest priority setting of 0, is not sent over the link by the Link Manager process 114 when it is received, it will instead be queued. Upon receipt of the second and third packet messages, with respective priority settings of 7 and 2 respectively, the Link Manager process 114 sends the second and third packet messages straight out over the low bandwidth link 120 to the receiving entity host computer. If the congestion on the low bandwidth link 120 had then eased and if the first packet message had not had a limited time-to-live associated with it, the

Link Manager process 114 might then have sent the first message over the low bandwidth link to the receiving entity host computer.

Had the Link Manager process 114 done so, it would have had the result that the latest positional update message received by the client would in fact have dated from one and two time periods respectively before the positional updates carried by the second and third messages which had already been received by the client. Instead, the time-to-live setting on the message will cause the message to be discarded from the Link Manager process queue 116 before it can be sent over the low bandwidth link 120.

It will be appreciated, as indicated above, that the longest delay, indeed the only significant delay, the packet message will experience before transmission, is that resulting from being queued. The time-to-live setting, apart from its advantage in the prevention of an erroneous re-ordering of packet messages as explained above, is also of more general assistance in queue memory management. The packet message queue 116 could be swept, for example, when the queue 116 reaches a certain predetermined length. Memory space otherwise taken up by timed-out packet messages may thereby be freed up. It would still be necessary, however, to check for timing-out on the sending of each packet. Alternatively, the queue 116 could be swept before it had reached this certain length. This would free up memory space otherwise taken by timed-out packets that might otherwise have lived on in a queue 116 shorter than the certain length.

The fruits of this approach to management of the low bandwidth link by the Link Manager process, combining cyclic priority settings with time-to-live settings are well demonstrated in and will be discussed having regard to, Figure 7 and Figure 8.

A simulation was created as follows. A 'Server' process was set up on a single PC, with a small number of initial entities, each of which possess position state. A 'Client' process was set up with an identical state on the same machine and the two were connected via a 28.8 Kbit/s UDP/IP link via a 'reflector' machine (simply bouncing packets from the server back to the client).

The Server moved all the entities in random directions at a constant rate and, at a fixed periodicity for each entity (and hence accordingly, the assigned time-to-live for the packet messages of each entity), each sent a position update to the client. The cyclic priority settings in this example used $n=3$, i.e. a four level priority scheme. The efficiency of the state transfer can then be measured by assessing the difference between the server and client state. With a perfect link (infinite bandwidth and zero latency) the states would be identical. With a non-perfect link the states will differ, and this difference can be measured by applying Pythagoras's rule to determine the distance between server and client entity positions.

This state difference for a given number of entities can be expressed as either.

1. An average of all the server-client position differences over a long period of time.
2. The average of the maximum server-client position differences over a long period of time.

As the number of entities is increased the volume of state updates generated is increased, and hence the greater the load imposed on the server to client link.

The approaches of sending with no priority and sending with a four level cyclic priority behave much the same with up to 25 entities sending data. This is because below 25 entities all the state updates generated within a fixed time period can be sent over the server to client link in the same period.

As the number of entities increases, and hence the volume of traffic sent to the client increases, the error detected for the non-prioritised case rapidly deteriorates. This is because the UDP link discards excess data, essentially making a random selection of the supplied updates. One entity might manage to transmit a sequence of 10 position updates successfully, whilst another manages to get none through over the same period.

In contrast, between 25 and 95 entities the cyclic priority scheme shows a large improvement in both the average and the maximum error detected. This is because

use of the cyclic priority ensures that when link bandwidth is exceeded each sending entity gets a fair sub-division of what is available. For example, when generated traffic is three times the available bandwidth (75 entities) the maximum error is around 4 times worse for the non-prioritised case.

5

When the volume of traffic generated equals the link bandwidth multiplied by the number of priority levels then the priority scheme approach degenerates to the same error level as the non-prioritised approach. This is because not enough bandwidth is available to carry all the top priority state updates, and hence which updates are
10 selected once again becomes a function of the network. This can be clearly seen on the maximum error graph, as the number of entities approaches 100.

A third embodiment according to the invention will now be discussed having regard to Figures 1, 2, 3, 9, 10 and 11. This third embodiment differs from the first and second
15 only in terms of the structure of the packet messages sent by the source entities and in the functionality of the Link Manager process, so appropriate reference should again be made to Figures 1, 2 and 3 where appropriate. Figure 9 illustrates a procedural flowchart of aspects of functioning of this second embodiment according to the invention. Again, the steps of the procedural flowchart will be carried out
20 through execution of the software application programs running on the link management computer.

According to this third embodiment and having particular regard to Figure 3, the source entities 100, 100' transmit packet messages 300 with the following structure.
25 The first field 306 in the packet header 302 will be set with the identity of the sending source entity. The second field 308 in the packet header 302, identifying a destination entity, need not be set as there exists only one destination entity in this example.

30 The third field 310, identifying a priority setting, will again be set as with the first and second embodiments. Each individual packet message is assigned a priority setting on a scale of for example, 0 (lowest priority) to n (highest priority). Again crucially, a sequence of n + 1 such messages is sent out from the source entity, not with a fixed

priority assigned to the source entity in question, but with cyclically varying priorities. The cyclic priorities vary such that if packets were discarded on a priority basis, the remaining packets in the sequence would be left as evenly spaced as possible.

- 5 Again, a simple example where $n = 7$, a corresponding suitable cyclic variation in priority is a sequence of eight messages with priorities set at 0, 7, 2, 4, 1, 6, 3 and 5 respectively. The 0, 7, 2, 4, 1, 6, 3, 5 sequence will ensure that as successive low priorities have to be dropped, a message with priority 0 being dropped first followed by a message with priority 1 and so on, the remaining packets will be as evenly
10 spaced as possible.

In this third embodiment the fourth field 312, giving a time-to-live period, will not be set. In this third embodiment however, the fifth field 314 in the packet header 302, giving an indication of the packet payload type, is set.

15

- It will be recalled from the above discussion of the second embodiment that the setting of a time-to-live period by a sending source entity corresponded with the periodicity of the source entity. The example given above had a source entity sending packet messages every 20 nominal temporal units with a corresponding time-to-live
20 of 20 nominal temporal units. Each packet message carrying a state update need not live longer than the source entity sending period since after another such period of time, a new state update packet message will have been sent.

- In this third embodiment a time-to-live period is not set. Instead each source entity
25 sends out state update messages with the fifth field 314 of the packet header 302 set with an identification of the packet payload type, which in this example will be a position indication. Each source entity will thus send out state update packet messages only when it is necessary.

- 30 According to this third embodiment, the Link Manager process 114 screens the incoming aperiodic packet messages, such that packet messages from the same source entity with the same payload type can be combined to make a new packet message with the latest possible state update.

Thus, streams of such packet messages, again bearing cyclic priority settings, as discussed above, are sent out from the source entities 100, 100'. As with the first embodiment, these packet messages are then passed, via the respective server
5 computers 104, 104', to the link management computer 110.

Having regard to Figure 9, in a first step 900, after these packet messages are received at the first and second link management computer input ports 112, 112', they are then passed to the Link Manager process 114.

10

In a second step 902, when a new packet message is received at one of the the link management computer input ports 112, 112' , the Link Manager process 114 will read the first field 306 of the packet header 302, to determine the identity of the sending source entity. It will be recalled that in this embodiment, the second field 308
15 in the packet header 302, identifying a destination entity, is not set as there exists only one destination entity in this example. In a third step 904, the Link Manager process 114 also reads the fifth field 314 in the packet header 302, giving an indication of the packet payload type, which in this example will be a position indication. In a fourth step 906, the Link Manager process 114 also reads the third
20 field 310 of the packet header 302 giving the assigned priority setting.

The Link Manager process 114 will again maintain a simple packet message queue 116, ordered on priority as with the first and second embodiment.

25 In a fifth step 908, the Link Manager process 114 will then sweep this packet queue 116, reading the first and fifth fields 306, 314 in the queued packet messages packet headers 302, to determine whether or not a packet message which has been received from that source entity and bearing that type of payload is currently being queued. Since it is assumed that packet messages sent from source entities arrive at the link
30 management computer 110 with no significant delay, it will be assumed that a later arriving packet message carries a more recent state update than an earlier arriving packet message.

It is to be noted that since this match is performed on, at least the sending source identity and the payload type, each sending source entity will be free to denote whatever payload type it requires with whichever payload type indicator is wished (i.e. source A might choose to denote a position payload with type=1, whereas
5 source B might denote a temperature payload, for example, with the same type type=1).

Accordingly, if the Link Manager process 114 locates such an earlier packet message, from the same source entity and bearing the same type of payload, which is still
10 queued, then in a sixth step 910, the Link Manager process will read the third field 310 of the queued packet message packet header 302 giving the assigned priority setting.

In a seventh step 912 the Link Manager process 114 may create a new packet
15 message as follows. Having regard to Figure 2, the Link Manager process will write into the first field 306 of the packet header 302 of the new packet message, the common source entity identity read from the first field 306 of the packet header 302 of both the newly received packet message and the queued packet message. The Link Manager process 114 will write into the payload section 304 of this new packet
20 message the payload of the packet message just received by the link management computer, in other words the most recent state update available. The Link Manager process 114 will compare the priority settings in the third field 310 of the packet headers 302 of the queued packet message and the newly received packet message and will write into the third field 310 of the packet header 302 of the new packet
25 message, the higher of the two priorities of the queued packet message and the newly received packet message.

In an eighth step 914, the Link Manager process 114, having created such a new packet message then sorts this new packet message into the simple priority order
30 queue 116.

In a ninth step 916, the Link Manager process 114 will discard the newly received packet message and the queued packet message.

By way of an alternative, it will be appreciated that instead of the seventh and eighth steps 912, 914, the process might be carried out as follows. If the queued packet message had a higher priority than the newly received packet, then the payload of the newly received packet message could simply be copied into the queued packet message payload with the newly received packet message then being discarded. If, however, the newly received packet had a higher priority than the queued packet message then the queued packet message could simply be discarded and the newly received packet message sorted into the priority queue 116.

10

Where the Link Manager process 114 does not find a match on the criteria then in a tenth step 918 the received packet message will simply be sorted directly into the queue 116.

- 15 With the third embodiment according to the invention, the Link Manager process 114 will again test the link 120 for capacity to send the packet message at the head of the priority queue 116 before sending, as with the first and second embodiments.

As with the first and second embodiments, for as long as the instantaneous total bandwidth of the packet message traffic received by the link management computer 110 has been less than the known (low) bandwidth of the communication link 120, all packet messages placed in the Link Manager process queue 116 will be sent out on the link 120 with no queuing delay. The receiving entity will receive all the packet messages sent from all the source entities. Again, as with the first and second

25 embodiments, as the amount of packet message traffic increases, there will come a point when the total bandwidth of the incoming packet message traffic exceeds the bandwidth of the communication link 120. At this point it will again no longer be possible to send on simultaneously all the packet messages intended for the receiving entity.

30

The third embodiment according to the invention will provide all the advantages of the first embodiment according to the invention in terms of the cyclic priority settings being able to deliver an optimal and graceful degradation of service as the total traffic

bandwidth exceeds the low-bandwidth link capacity. The third embodiment according to the invention will, however, provide yet further advantages in terms of performance over the first embodiment, as will now be explained.

- 5 The fruits of the approach to management of the low bandwidth link by the Link Manager process as discussed having regard to this third embodiment according to the invention are well demonstrated in and will be discussed having regard to, Figures 10 and 11.
- 10 A similar simulation to that discussed having regard to Figures 7 and 8 was conducted. This experiment shows the results of using no priority scheme at all, a cyclic priority scheme on its own, and a cyclic priority scheme with a message combination mechanism. In this example, a cyclic priority scheme with $n=7$, i.e. an eight level priority scheme, was used
- 15 Not using any priority scheme performs adequately up to the point where the volume of traffic generated on the server exceeds the bandwidth available for updating the client (30 entities). From this point on both the average and maximum error increase as state updates are discarded at random by the network.
- 20 The result of using an 8 level priority scheme on its own is shown as a dotted line on both graphs. This performs significantly better than the none-prioritised case at some traffic volumes (e.g. 70 entities and 90 entities), and significantly worse at other traffic volumes (e.g. 35 entities). Each of the peaks in error from this data set
- 25 correspond to when the traffic volume generated is equal to some integer multiple of the limited volume that can be carried between server and client. As discussed in an example above, the sequence of events that cause these peaks is:
1. At time t Entity X generates state update S_t with priority P .
 - 30 2. S_t is queued because the link is currently fully utilised by higher priority updates.
 3. At time $t + 1$ Entity X generates state update with S_{t+1} with priority $>P$.
 4. S_{t+1} is sent because it has a high enough priority to supplant any other queued traffic.

5. Fluctuation in the generation of new state updates allows lower priority updates that have been queued to be sent, S_t is sent, and the receiving entity on the client treats this as update $S_t + 2$.

5 Hence these error peaks are only generated when fluctuations in the rate of state update generation can cause the above re-ordering case, i.e. when the traffic volume generated is close to some integer multiple of the available bandwidth, and one particular priority level is only occasionally making it onto the low bandwidth link.

10 This problem is resolved by the third embodiment according to the invention, shown as the dashed line on both graphs, wherein new messages are combined with any old queued messages from the same entity. Hence, the type of re-ordering problem outlined above is eliminated, and the scheme performs well at all traffic levels.

15 Having regard to Figure 3, and in particular the second field 308 of the packet header 302, it will be appreciated that it is possible to send to multiple clients from a single link manager if respective different destination entity identities are provided. The Link Manager process may then maintain multiple sorted priority packet message queues, on the basis of, for example, one such queue per client.

20

Thus the destination entity identity field may be used by the link manager to determine which queue any incoming packets should be placed in. Having used the packets

destination field to determine the relevant queue, the link manager may then utilise
25 the mechanisms previously described to place the packet in the queue and then either deliver or discard it. It will be appreciated that once the link manager supports multiple clients, and hence multiple priority sorted packet queues, it will need to round-robin between

these queues, allowing each of them processing time to send any packets from the
30 head of their queue that the link to their client may carry at that moment in time.

Figure 12 illustrates a fourth embodiment according to the invention.

Each client 1200, 1202, 1204 is treated as a single receiving entity 1206, and has zero or more sending entities 1208 resident on it. All clients are connected to a link manager 1210. Each client 1200, 1202, 1204 can generate any number of updates from any of its sending entities, and address these to any of the other client
5 receiving entities. Each update is sent to the link manager 1210, and this has the task of distributing the incoming updates to the relevant clients 1200, 1202, 1204. Since any single client 1200, 1202, 1204 may be receiving updates from numerous other clients 1200, 1202, 1204 a fan-in type problem occurs, whereby each client is only generating a limited number of updates, but particular clients are being sent a larger
10 number of updates than the bandwidth of their network connection allows.

To handle this problem effectively the link manager 1210 maintains a priority sorted queue 1212, 1214, 1216 for each client, and this queue 1212, 1214, 1216 is emptied down to the client 1200, 1202, 1204 at whatever bandwidth the link
15 manager to client link can support. These queues 1212, 1214, 1216 are managed using the previously described cyclic priority scheme, whereby each update is allocated a priority by the client as its generated, together with either a time to live or a payload type. Hence, if any client is being sent more updates than its link bandwidth will allow it to receive the link manager will act to ensure that graceful
20 degradation in the periodicity of the updates received occurs.

This approach assumes that the upstream traffic generated from client to link manager will not exceed the available link bandwidth. i.e. that the client can regulate the periodicity of updates generated by the sending entities to ensure the total traffic
25 generated in any single time period remains below the maximum it can transmit within that same period. If this is not the case then an identical scheme to that employed at the link manager can be adopted for use at the client i.e. a priority sorted queue, messages sent from its head at whatever rate the link bandwidth allows, and with a time to live or data combination approach used to constrain the queue size and
30 prevent message re-ordering.

In this implementation each message would pass through up to two queues, one before leaving the client, and one at the link manager whilst awaiting delivery to a

client. Note that if using a time to live approach with this double queuing scheme, then the time to live in the message must be correctly adjusted before a message is sent. For example, if a message is generated at $t=0$ with a TTL of 20, and remains in the client-side queue for 5 time periods, then it should be sent up to the link manager
5 with a TTL of 15 (i.e. Original TTL minus the time spent in the queue).

Whilst the above embodiments have illustrated techniques where a priority label is assigned to each packet message by a source, this need not always be the case. So long as the sequence order in which packet messages are transmitted by a source is
10 preserved (for example, a sequence order field), an allocation of the priority labels could be performed downstream of the sources. Indeed, given that priority labels are allocated for the purposes of making decisions about which part of the packet message queue to sort a received packet message into, the sorting algorithm (utilising the cyclic label sequence technique as above and with knowledge of the sequence
15 position of a given received packet message from a given source) might be used to sort a received packet message into the same portion of the queue as would have been performed in two steps as above, with an assignment of an appropriate priority label and a priority based sorting into a queue.

20

25

CLAIMS

1. A packet message source comprising:

5 means arranged to include a respective packet message payload in each packet message of a sequence of packet messages;

means arranged to associate a priority label with each successive packet message in said sequence in accordance with a predetermined cyclic sequence of such labels;

10 said priority labels each representing one of a plurality of priority levels and the position of each label in the cyclic sequence being such that the number of consecutive lower priority labels between that label and the nearest label of equal or higher priority is substantially maximised; and

15 means arranged to send such packet messages.

2. A packet message source as claimed in claim 1, wherein said packet message source has an associated dynamic state and said respective packet message
20 payload comprises a source state update message.

3. A packet message source as claimed in claim 1 or claim 2 further comprising:

25 means arranged to associate a time-to-live setting with each packet message.

4. A packet message source as claimed any one of claims 1 to 3 further comprising:

30

means arranged to associate a packet message source identity with each packet message; and

means arranged to associate a packet message payload type setting with each packet message.

5 5. A packet messaging system comprising:

a plurality of packet message sources according to any one of claims 1 to 4 ;and

a communications link interface;

10

said communications link interface comprising:

an input port for receiving packet messages from said plurality of packet message sources;

15

means arranged to read a priority label associated with each received packet message;

a queue for queuing received packet messages in order of their associated priority labels; and

20

an output port for sending each packet message at the head of said queue onto a communications link.

25 6. A packet messaging system according to claim 5 when depending on claim 3, said communications link interface further comprising:

means arranged to read a packet message time-to-live setting associated with each received packet message;

30

means arranged to associate with each respective packet message an indication of the period of time that packet message has been queued ;and

means arranged to discard each packet message whose associated indication indicates that that packet message has been queued for a period of time longer than the associated packet message time-to-live setting.

- 5 7. A packet messaging system according to claim 5 when depending on claim 3, said communications link interface further comprising:

a clock; and

- 10 means arranged to replace the packet message time-to-live setting associated with each packet message with an adjusted time-to-live setting resulting from the addition of the time-to-live setting read from the packet message and the local time at which the packet was received;

- 15 and in which the means arranged to associate an indication of a period of time with each packet message, associates with at least one packet message, the result of said adjusted time-to-live read from each said packet message minus the local time at which said packet message arrived such that the means arranged to discard each packet message are so arranged to discard a packet message if said associated sum
20 is less than or equal to zero.

8. A packet messaging system according to claim 5 when depending on claim 4, said communications link interface further comprising:

- 25 means arranged to read a packet message source identity from each received packet message;

means arranged to read a packet message payload type setting associated with each received packet message;

30

means arranged to test the packet message queue for a queued packet message with an associated source identity matching that of the received packet message;

means arranged to read the priority label associated with the matching queued packet message;

means arranged to sort into the queue on the basis of priority label a packet message replacing the matched received and queued packet messages, having the associated source identity of the matched received and queued packet messages, the payload of the received packet message and whichever of the associated priority label of the matched received and queued packet message represents the higher priority.

10 9. A packet messaging system comprising:

a plurality of packet message sources each comprising:

means arranged to include a respective packet message payload in each packet
15 message of a sequence of packet messages; and

means arranged to send such packet messages; and

a communications link interface comprising:

20

an input port for receiving packet messages from said plurality of packet message sources;

a queue for queuing packet messages in order of respective allocated priority labels; successive packets, considered in said sequence at each one of said plurality of
25 packet message sources, having been allocated said priority labels in accordance with a predetermined cyclic sequence of such labels; said priority labels each representing one of a plurality of priority levels and the position of each label in the cyclic sequence being such that the number of consecutive lower priority labels between it
30 and the nearest label of equal or higher priority is substantially maximised; and

an output port for sending each packet message at the head of said queue onto a communications link.

10. A method of packet messaging comprising:

including a respective packet message payload in each packet message of a sequence
5 of packet messages;

associating a priority label with each successive packet message of said sequence in accordance with a predetermined cyclic sequence of such labels; said priority labels each representing one of a plurality of priority levels and the position of each label in
10 the cyclic sequence being such that the number of consecutive lower priority labels between said label and the nearest label of equal or higher priority is substantially maximised; and

sending such packet messages.

15

11. A method of operating a packet messaging system including a plurality of packet message sources and a communications link interface, said method comprising:

20 allocating successive packets, considered in their original sequence at each one of the packet message sources, priority labels in accordance with a predetermined cyclic sequence of such labels; said priority labels each representing one of a plurality of priority levels and the position of each label in the cyclic sequence being such that the number of consecutive lower priority labels between it and the nearest label of
25 equal or higher priority is substantially maximised;

queuing said packet messages in a queue in order of respective allocated priority labels at said communications link interface; and

30 sending each packet message at the head of said queue onto a communications link.

12. A computer readable storage medium, said medium embodying computer readable code for reading into a computer and executable by said computer to perform the method of claim 10 or claim 11.

5 13. A Link Manager for managing the sending of packet messages on a link;

the packet messages originating from a plurality of packet message sources; each packet message source having an associated state; each packet message including an associated packet message priority setting on a scale of n to m and a packet message source state update;
10 each packet message having been sent by a source sending sequences of $m - n + 1$ packet messages cycling through each priority setting of said priority setting scale such that successively dropping packet messages from the sequence on a priority basis leaves the remaining packet
15 messages of the sequence as evenly spaced as possible;

the Link Manager comprising:

at least one packet message input port to receive such packet messages;
20 means arranged to read the priority setting associated with each such received packet message

means arranged to sort each such received packet message into a packet message
25 queue on the basis of the priority setting associated with each such received packet message;

means arranged to test the link for sufficient capacity to send the packet message at the head of the queue; and

30 means arranged to send the packet message at the head of the queue, when the link has sufficient capacity, out through at least one output port onto the link.

14. A method of managing the sending of packet messages from a plurality of sources on a link;

5 the packet messages originating from a plurality of packet message sources;
each packet message source having an associated state; each packet message
including an associated packet message priority setting on a scale of n to m
and a packet message source state update;
each packet message having been sent by a source sending sequences of $m -$
 $n + 1$ packet messages cycling through each priority setting of said priority
10 setting scale such that successively dropping packet messages from the
sequence on a priority basis leaves the remaining packet
messages of the sequence as evenly spaced as possible;

the method comprising:

15

receiving such packet messages at at least one input port of a link manager;

reading the priority setting associated with each such received packet message;

20 sorting each such received packet message into a packet message queue on the
basis of the priority setting associated with each such received packet message;
and

sending the packet message at the head of the queue out through at least one
25 output port of the link manager onto the link.

1/9

Fig.1.

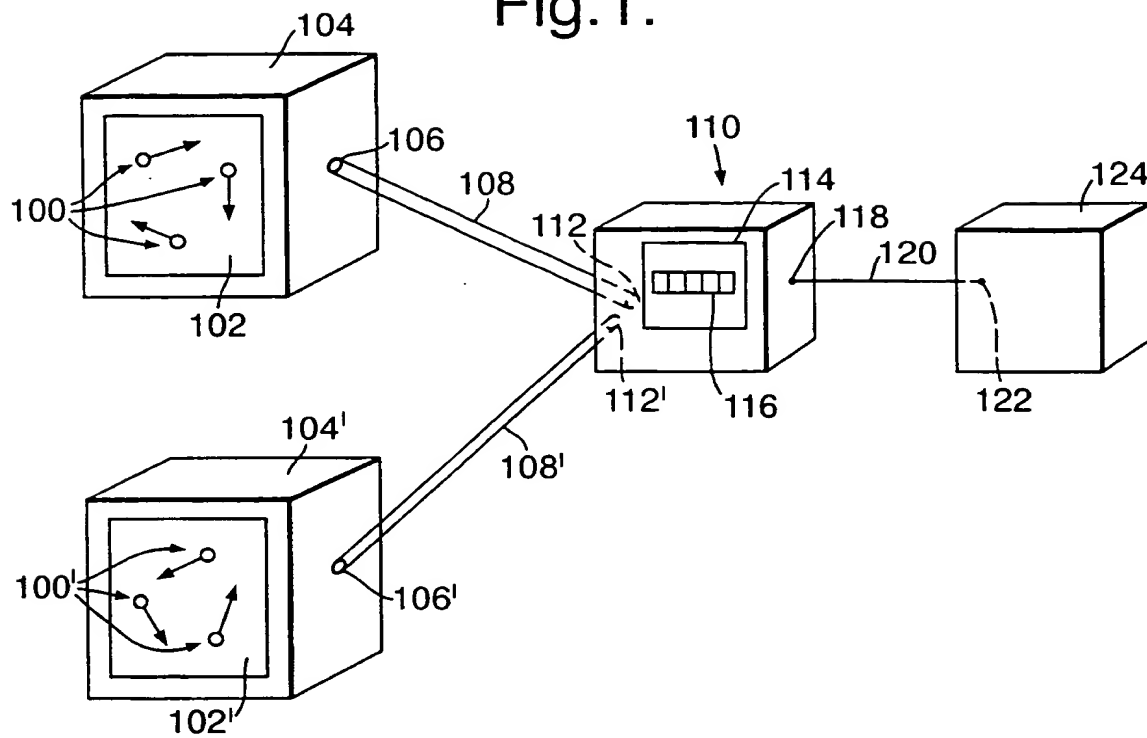


Fig.2.

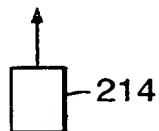
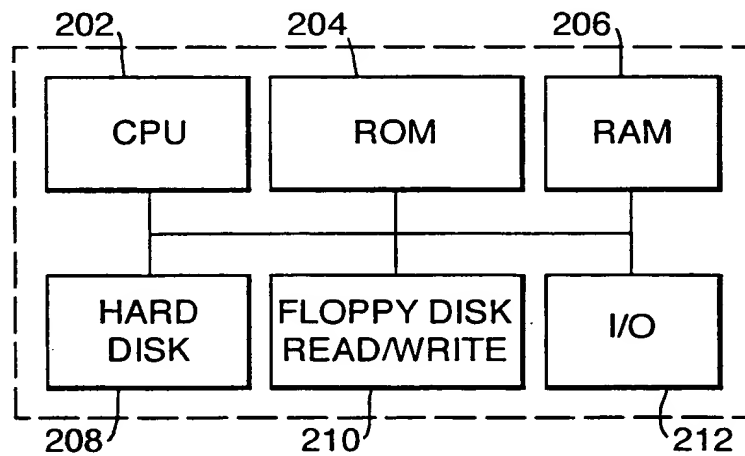


Fig.3.

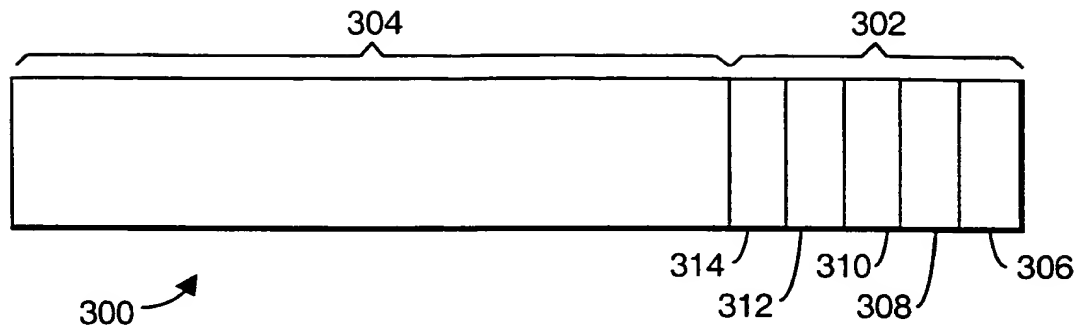
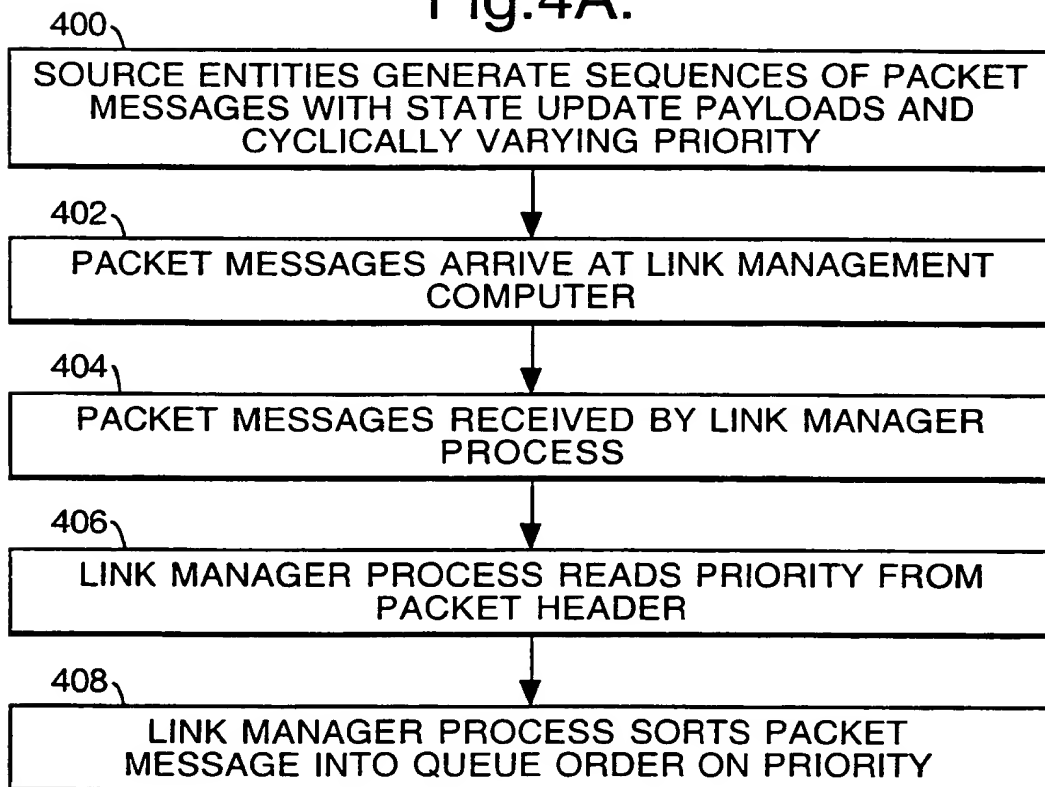


Fig.4A.



3/9

Fig.4B.

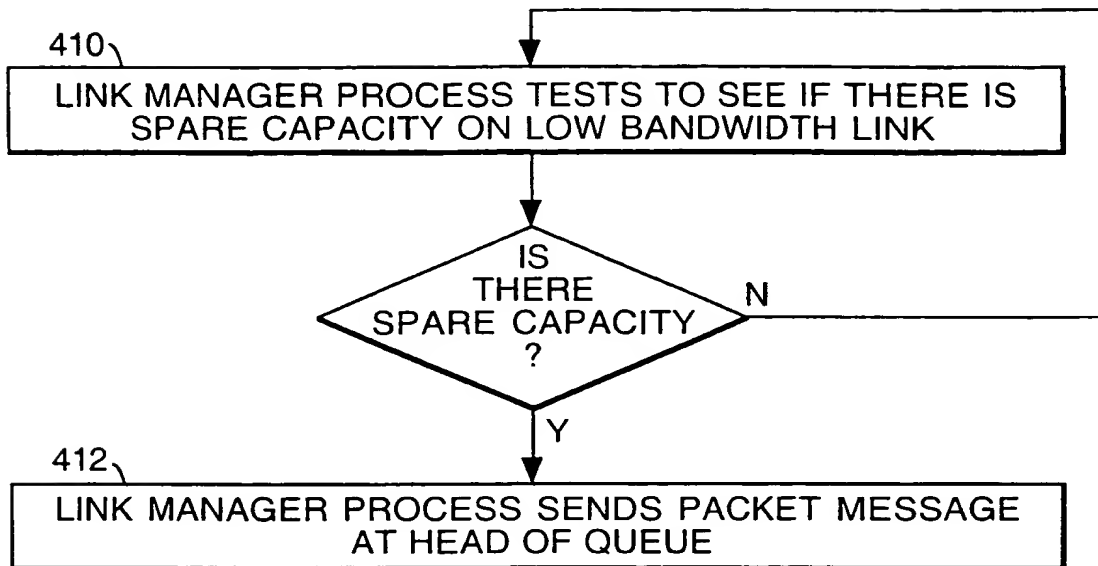
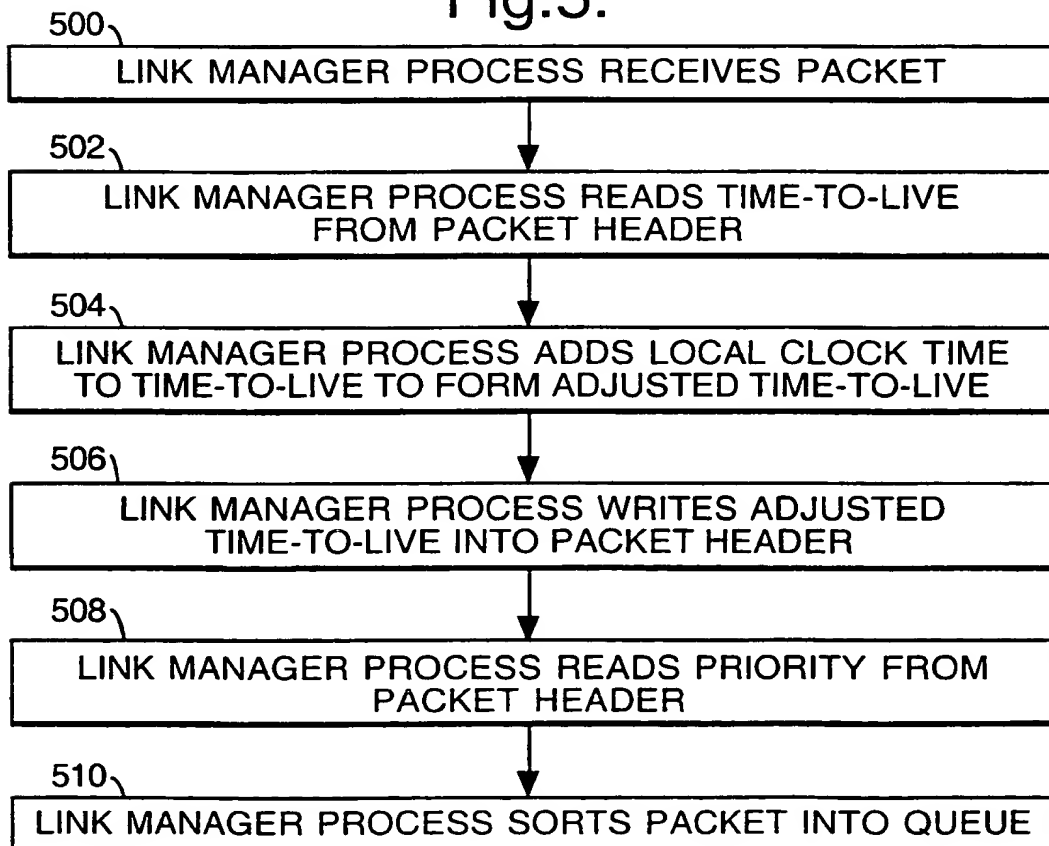


Fig.5.



4/9

Fig.6.

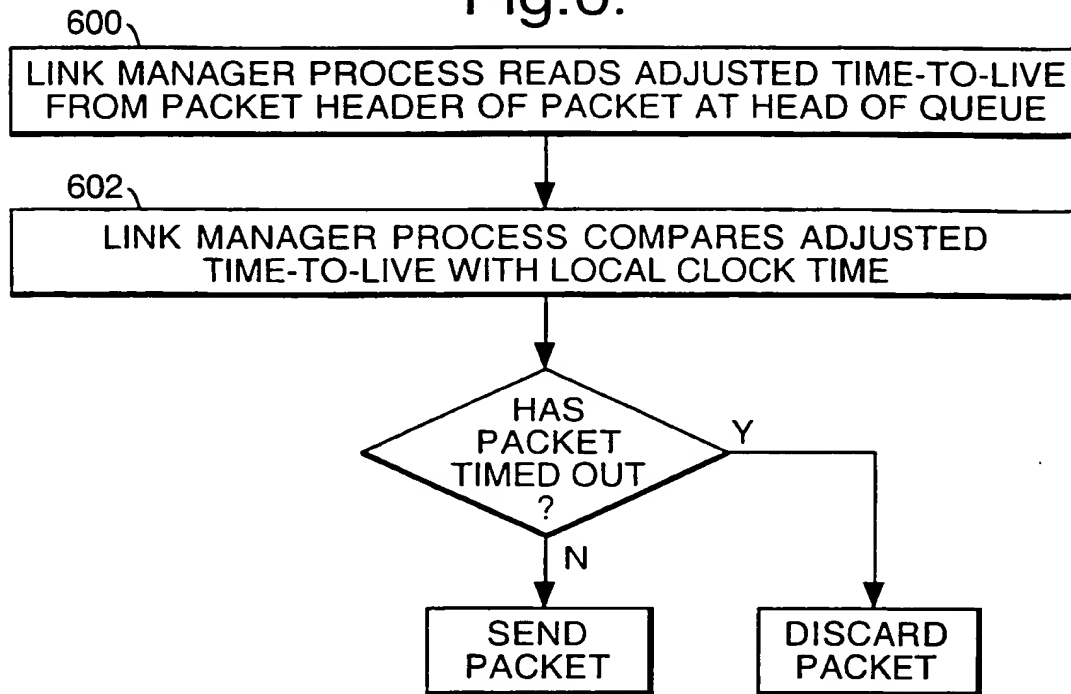


Fig.7.

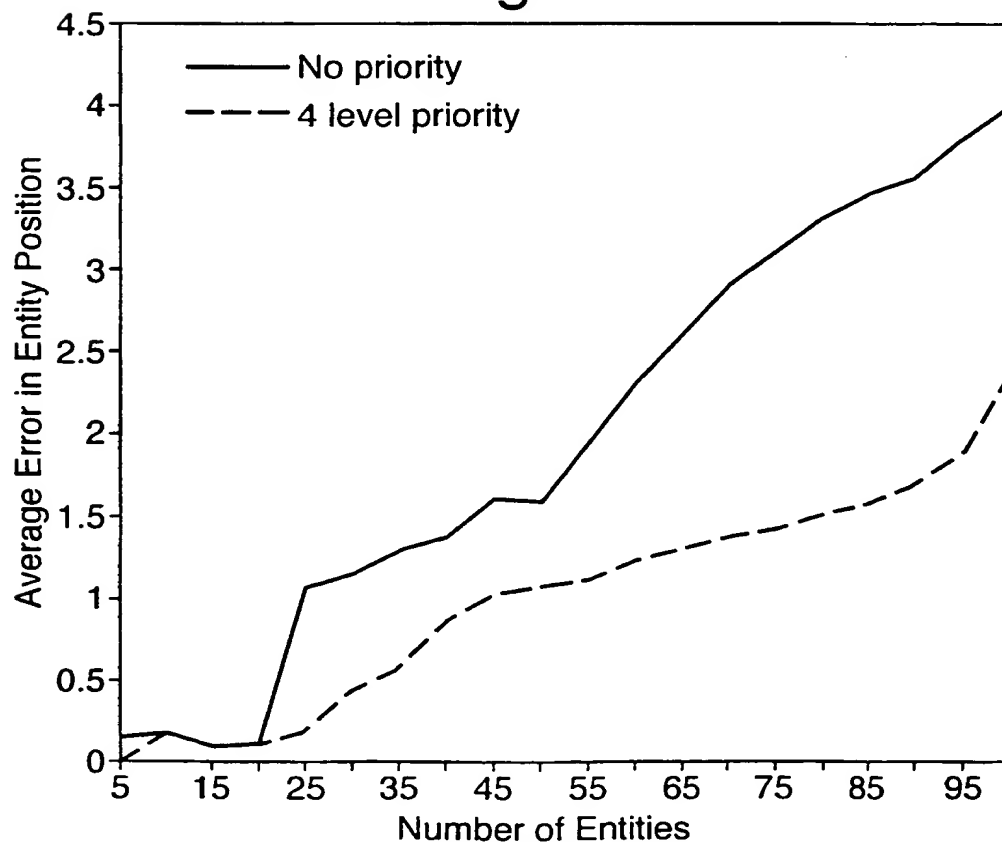


Fig.8.

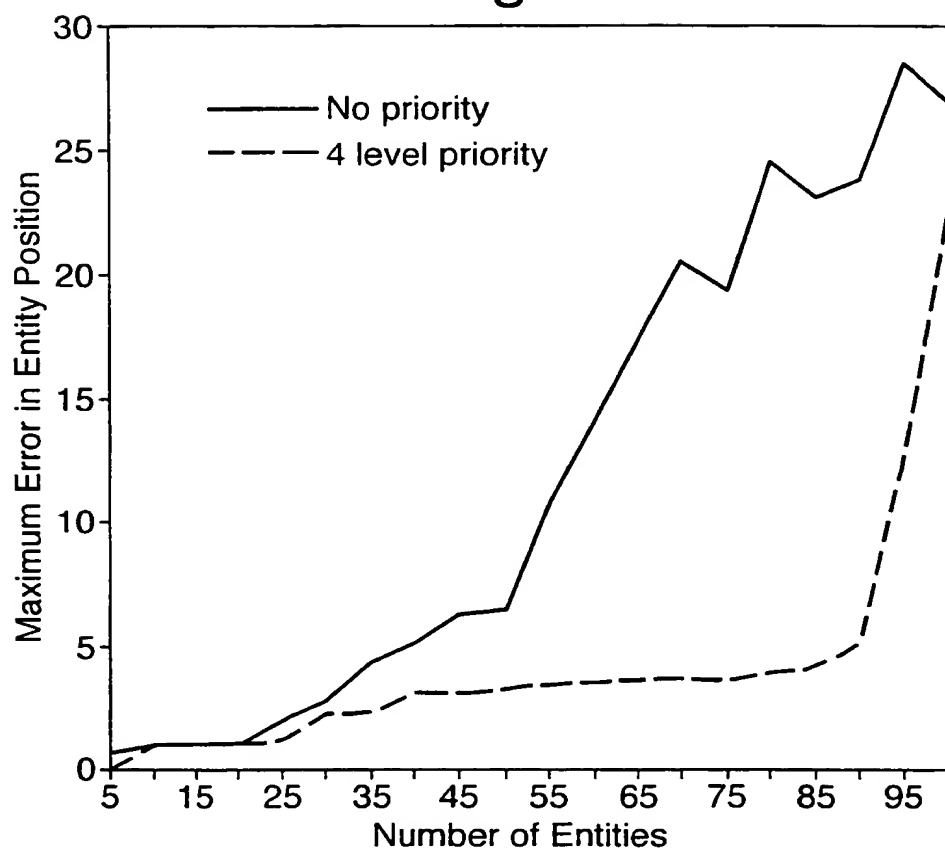


Fig.9A.

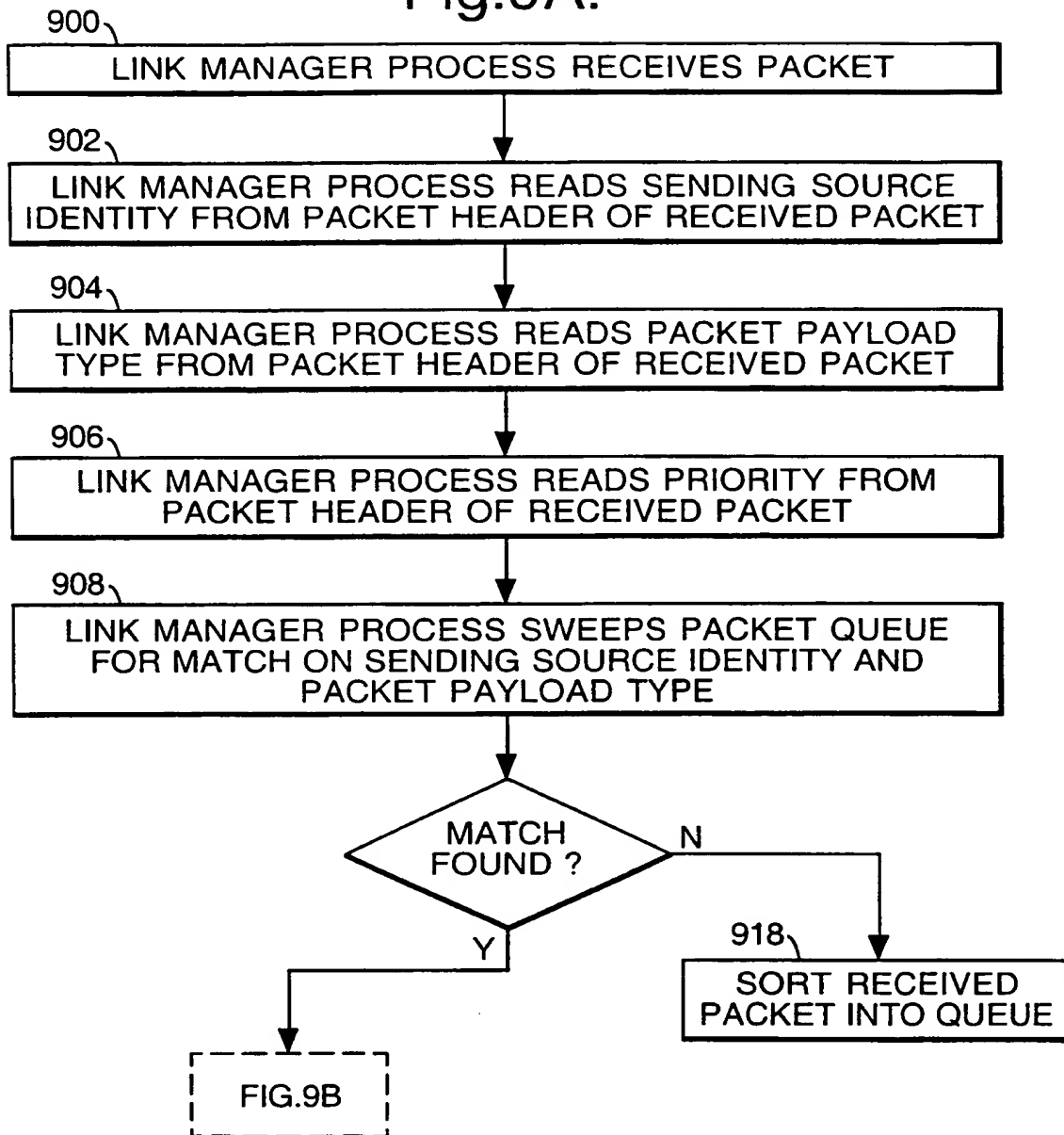


Fig.9B.

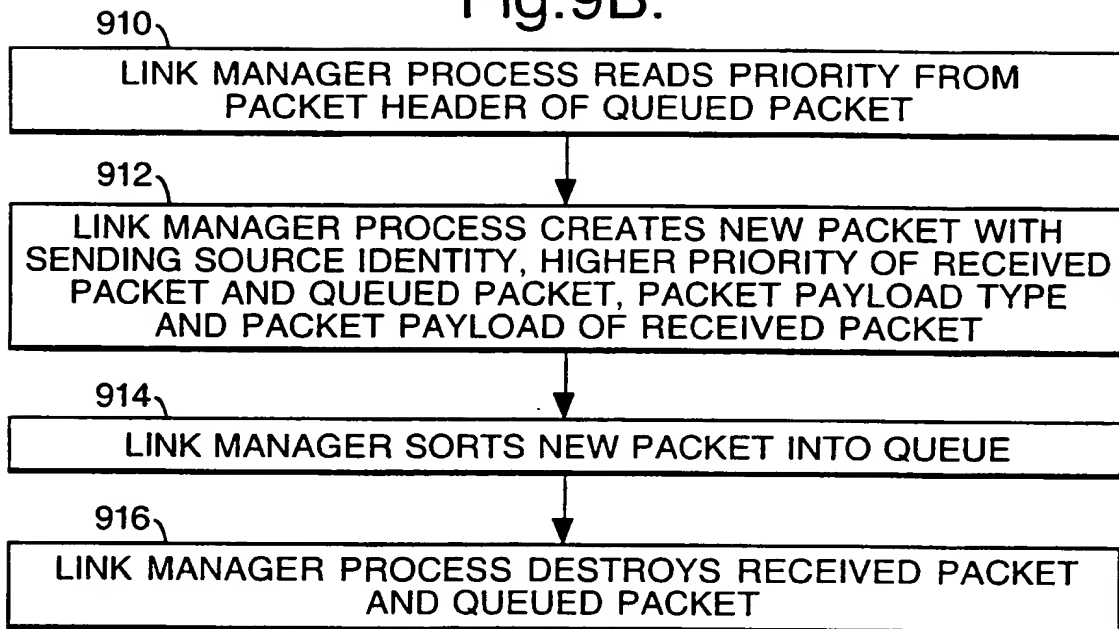


Fig.10. 8/9

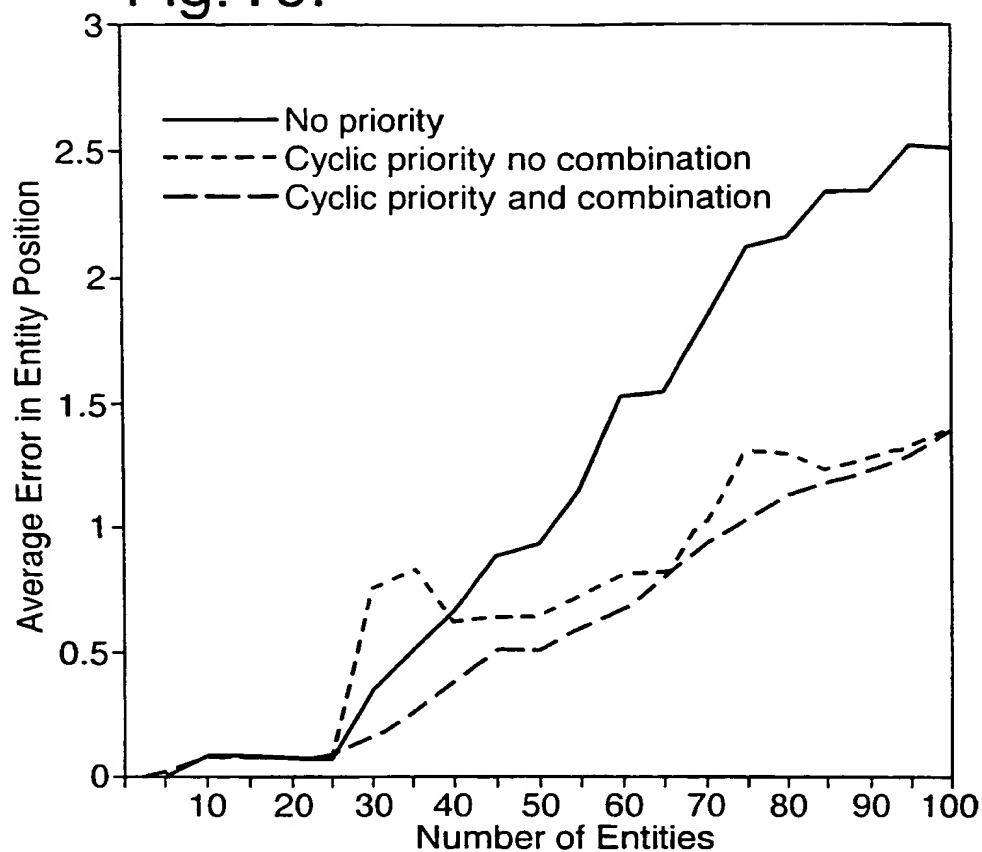


Fig.11.

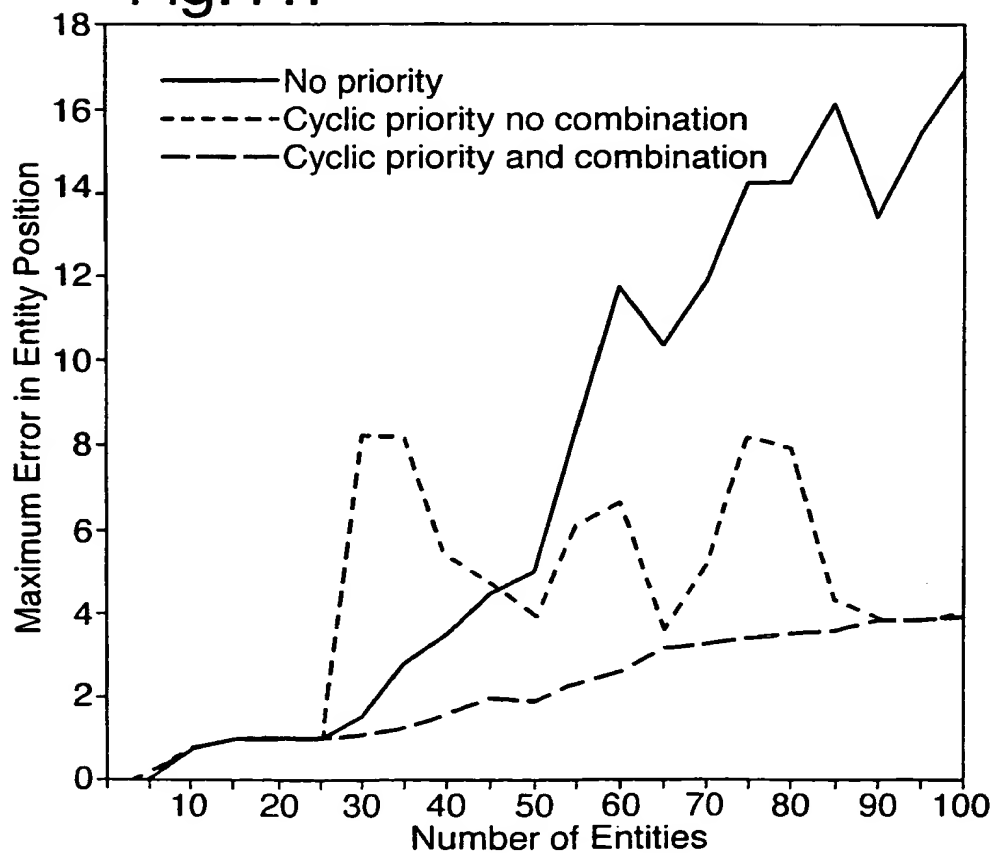
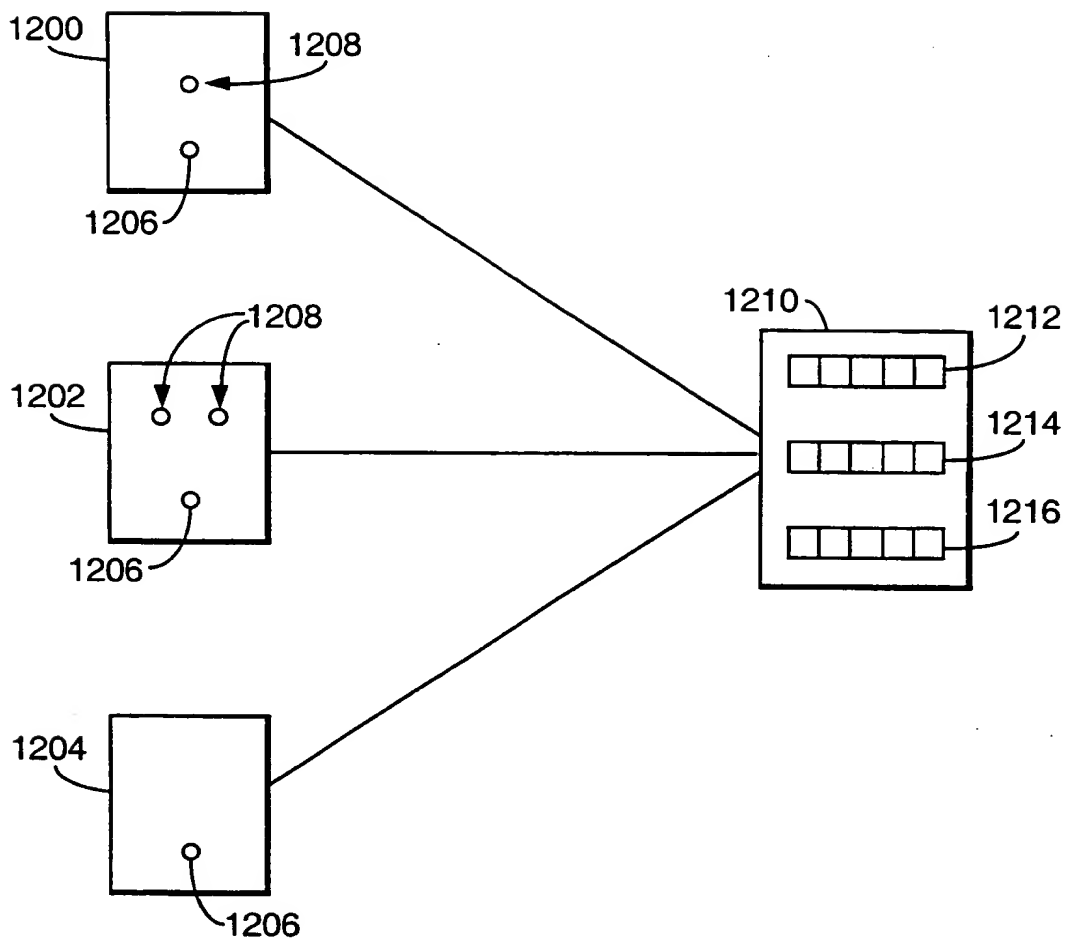


Fig.12.



INTERNATIONAL SEARCH REPORT

International Application No

PCT/GB 00/01135

A. CLASSIFICATION OF SUBJECT MATTER

IPC 7 H04L29/06

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 7 H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, WPI Data, PAJ, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WO 97 43846 A (KOUTS MICHAEL ;MCKELLER DAVID (CA); PLUMB MARC (CA); SHIELDS DANIE) 20 November 1997 (1997-11-20) page 9, line 1 -page 13, line 31	1,9-11, 13,14
A	FUNKHOUSER T A: "RING: A CLIENT-SERVER SYSTEM FOR MULTI-USER VIRTUAL ENVIRONMENTS" PROCEEDINGS OF THE SYMPOSIUM ON INTERACTIVE 3D GRAPHICS,US,NEW YORK, ACM, page 85-92 XP000546194 ISBN: 0-89791-736-7 page 87, left-hand column, paragraph 2 -page 88, right-hand column, paragraph 6 page 91, right-hand column, paragraph 2 --- -/-	1,9-11, 13,14

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search

8 September 2000

Date of mailing of the international search report

18/09/2000

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Authorized officer

RAMIREZ DE AREL., F

INTERNATIONAL SEARCH REPORT

International Application No

PCT/GB 00/01135

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5 659 691 A (PRETTEGIANI JERRY ET AL) 19 August 1997 (1997-08-19) column 1, line 45 -column 2, line 14	1,9-11, 13,14

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/GB 00/01135

Patent document cited in search report		Publication date	Patent family member(s)		Publication date
WO 9743846	A	20-11-1997	AU	2687697 A	05-12-1997
			EP	0958687 A	24-11-1999
US 5659691	A	19-08-1997	AU	687888 B	05-03-1998
			AU	7736694 A	10-04-1995
			CA	2172535 A	30-03-1995
			WO	9508793 A	30-03-1995
			EP	0721614 A	17-07-1996
			JP	9503082 T	25-03-1997
			NZ	273893 A	27-05-1998
			NZ	330003 A	28-10-1999
			US	5950202 A	07-09-1999

PATENT COOPERATION TREATY

PCT

INTERNATIONAL PRELIMINARY EXAMINATION REPORT

(PCT Article 36 and Rule 70)

REC'D 26 JUN 2001

PCT

Applicant's or agent's file reference A25771 WO	FOR FURTHER ACTION See Notification of Transmittal of International Preliminary Examination Report (Form PCT/IPEA/416)	
International application No. PCT/GB00/01135	International filing date (day/month/year) 24/03/2000	Priority date (day/month/year) 31/03/1999
International Patent Classification (IPC) or national classification and IPC H04L29/06		
Applicant BRITISH TELECOMMUNICATIONS PUBLIC LIM. COMP.et al.		

1. This international preliminary examination report has been prepared by this International Preliminary Examining Authority and is transmitted to the applicant according to Article 36.

2. This REPORT consists of a total of 13 sheets, including this cover sheet.


☒ This report is also accompanied by ANNEXES, i.e. sheets of the description, claims and/or drawings which have been amended and are the basis for this report and/or sheets containing rectifications made before this Authority (see Rule 70.16 and Section 607 of the Administrative Instructions under the PCT).

These annexes consist of a total of 14 sheets.

+ 2 annexes (Act 1 and 2)

3. This report contains indications relating to the following items:

- I ☒ Basis of the report
- II ☐ Priority
- III ☒ Non-establishment of opinion with regard to novelty, inventive step and industrial applicability
- IV ☒ Lack of unity of invention
- V ☐ Reasoned statement under Article 35(2) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement
- VI ☐ Certain documents cited
- VII ☒ Certain defects in the international application
- VIII ☒ Certain observations on the international application

Date of submission of the demand 20/10/2000	Date of completion of this report 21.06.2001
Name and mailing address of the international preliminary examining authority:  European Patent Office D-80298 Munich Tel. +49 89 2399 - 0 Tx: 523656 epmu d Fax: +49 89 2399 - 4465	Authorized officer Keller, M Telephone No. +49 89 2399 8807



**INTERNATIONAL PRELIMINARY
EXAMINATION REPORT**

International application No. PCT/GB00/01135

I. Basis of the report

1. With regard to the **elements** of the international application (*Replacement sheets which have been furnished to the receiving Office in response to an invitation under Article 14 are referred to in this report as "originally filed" and are not annexed to this report since they do not contain amendments (Rules 70.16 and 70.17)*):

Description, pages:

1,2,4-6,8-13,17, as originally filed
19-28

3,7,14-16,18 as received on 18/05/2001 with letter of 17/05/2001

Claims, No.:

1-17 as received on 18/05/2001 with letter of 17/05/2001

Drawings, sheets:

1/9-9/9 as originally filed

2. With regard to the **language**, all the elements marked above were available or furnished to this Authority in the language in which the international application was filed, unless otherwise indicated under this item.

These elements were available or furnished to this Authority in the following language: , which is:

- ☐ the language of a translation furnished for the purposes of the international search (under Rule 23.1(b)).
- ☐ the language of publication of the international application (under Rule 48.3(b)).
- ☐ the language of a translation furnished for the purposes of international preliminary examination (under Rule 55.2 and/or 55.3).

3. With regard to any **nucleotide and/or amino acid sequence** disclosed in the international application, the international preliminary examination was carried out on the basis of the sequence listing:

- ☐ contained in the international application in written form.
- ☐ filed together with the international application in computer readable form.
- ☐ furnished subsequently to this Authority in written form.
- ☐ furnished subsequently to this Authority in computer readable form.
- ☐ The statement that the subsequently furnished written sequence listing does not go beyond the disclosure in the international application as filed has been furnished.
- ☐ The statement that the information recorded in computer readable form is identical to the written sequence listing has been furnished.

4. The amendments have resulted in the cancellation of:

**INTERNATIONAL PRELIMINARY
EXAMINATION REPORT**

International application No. PCT/GB00/01135

- ☐ the description, pages:
☐ the claims, Nos.:
☐ the drawings, sheets:

5. ☐ This report has been established as if (some of) the amendments had not been made, since they have been considered to go beyond the disclosure as filed (Rule 70.2(c)):

(Any replacement sheet containing such amendments must be referred to under item 1 and annexed to this report.)

6. Additional observations, if necessary:

III. Non-establishment of opinion with regard to novelty, inventive step and industrial applicability

1. The questions whether the claimed invention appears to be novel, to involve an inventive step (to be non-obvious), or to be industrially applicable have not been examined in respect of:

- ☐ the entire international application.
☒ claims Nos. 1-17.

because:

- ☒ the said international application, or the said claims Nos. 15-17 relate to the following subject matter which does not require an international preliminary examination (*specify*):
see separate sheet
- ☒ the description, claims or drawings (*indicate particular elements below*) or said claims Nos. 1-14 are so unclear that no meaningful opinion could be formed (*specify*):
see separate sheet
- ☐ the claims, or said claims Nos. are so inadequately supported by the description that no meaningful opinion could be formed.
- ☒ no international search report has been established for the said claims Nos. 15-17.

2. A meaningful international preliminary examination cannot be carried out due to the failure of the nucleotide and/or amino acid sequence listing to comply with the standard provided for in Annex C of the Administrative Instructions:

- ☐ the written form has not been furnished or does not comply with the standard.
☐ the computer readable form has not been furnished or does not comply with the standard.

IV. Lack of unity of invention

1. In response to the invitation to restrict or pay additional fees the applicant has:

INTERNATIONAL PRELIMINARY EXAMINATION REPORT

International application No. PCT/GB00/01135

- ☐ restricted the claims.
 - ☐ paid additional fees.
 - ☐ paid additional fees under protest.
 - ☐ neither restricted nor paid additional fees.
2. ☒ This Authority found that the requirement of unity of invention is not complied and chose, according to Rule 68.1, not to invite the applicant to restrict or pay additional fees.
3. This Authority considers that the requirement of unity of invention in accordance with Rules 13.1, 13.2 and 13.3 is
- ☐ complied with.
 - ☒ not complied with for the following reasons:
see separate sheet
4. Consequently, the following parts of the international application were the subject of international preliminary examination in establishing this report:
- ☒ all parts.
 - ☐ the parts relating to claims Nos. .

VII. Certain defects in the international application

The following defects in the form or contents of the international application have been noted:
see separate sheet

VIII. Certain observations on the international application

The following observations on the clarity of the claims, description, and drawings or on the question whether the claims are fully supported by the description, are made:
see separate sheet

**INTERNATIONAL PRELIMINARY
EXAMINATION REPORT - SEPARATE SHEET**

International application No. PCT/GB00/01135

With respect to SECTION III:

- 1a. It is not considered appropriate to analyse in detail all the separate independent claims, since in the view of the comments in sections IV and VIII of this opinion, the number of independent claims will in any case have to be reduced.

Due to the multiplicity of independent claims (9 independent claims) directed to

- a packet message source (Claim 1),
- a packet messaging system (Claim 5),
- a packet messaging system (Claim 9),
- a method of packet messaging (Claim 10),
- a method of operating a packet messaging system (Claim 11),
- a computer readable storage medium (Claim 12),
- a Link Manager (Claim 13), and
- a method of managing the sending of packet messages (Claim 14),
- a signal (Claim 15)

it is totally unclear for which subject-matter protection is really sought. Therefore, the requirements of Article 6 PCT are not met.

- 1b. The Applicant has emphasized in his letter of 17.05.2001 that "claim 5 is dependent on claims 1 to 4 and is not an independent claim". This consideration is not shared by the examiner. The scope of protection of Claim 1 is directed to "a packet message source". Claim 5 has a totally different scope of protection which is "a packet messaging system". Moreover, the wording of Claim 5 itself does not contain an explicit dependance on any one of claims 1 to 4. Claim 5 merely defines a system which uses a plurality of packet message sources according to any one of claims 1 to 4. Finally, the requirements of Rule 6.4 PCT are not met. A dependent claim is any claim which includes ~~all~~ the features of one or more other claims **and** then states the additional features claimed. Although Claim 5 includes all features of the packet message *sources* according to any one of claims 1 to 4, Claim 5 does not specify additional features of the packet messages *sources* but parts of the system; these parts are external to the packet message source and thus are not additional features of the packet message source. Hence, Claim 5 is not a dependent claim in the sense of Rule 6.4 PCT.

**INTERNATIONAL PRELIMINARY
EXAMINATION REPORT - SEPARATE SHEET**

International application No. PCT/GB00/01135

1c.1 The Applicant has newly added Claims 15 to 17 directed to a *signal* per se.

These claims 15 to 17 are excluded from examination for the following reasons:

Firstly, Claims 15 to 17 fall under the exception of Rule 67.1 (v) PCT. Any presentation of information characterized solely by the content of the information is excluded under Rule 67 PCT. This applies, whether the claim is directed to

- the presentation of the information per se, as e.g. for (electrical) signals [immaterial entity],
- information recorded on a carrier, or
- processes and apparatus for presenting information.

1c.1.1 There **could** be patentable subject-matter in

- the information carrier, or
- the process, or
- apparatus

for presenting the information, if the presentation of information has new technical features. The arrangement (apparatus claim) or manner of representation (process claim), as distinguished from the *information content*, may well constitute a patentable technical feature.

(PCT-Guidelines; PCT Gazette, Section IV, IV-2.4 (e))

Hence, Claims 15 to 17 which are directed to signals per se (information content) are excluded under Rule 67.1 (v) PCT from examination.

1c.2 Secondly, Claims 15 to 17 have not been searched. According to the PCT-Guidelines (PCT Gazette, Section IV, VI-5.4 and VI-5.1) no examination is required for these claims.

1c.3 Thirdly, Claims 15 to 17 do not met the requirements of Article 6 PCT with respect to clarity, in particular clarity of category. The requirement that the claims shall be clear applies to individual claims and also to the claim as a whole. The clarity of the claims is of utmost importance for the purposes of formulating an opinion on the questions whether the claimed inventions appears to be novel, inventive and industrial applicable. Each claim shall be clear so that the meaning is clear from the wording of the claim **alone** (PCT Gazette, Section IV, III-4.1 and III-4.2). With respect to the category, there are only two basic kind of claims, viz. claims to a

physical entity (product; apparatus) and claims to an activity (process; use) [PCT Gazette, Section IV, III-3.1]. A signal, i.e. a presentation of information, is neither a physical entity nor an activity. Hence, a claim directed to a signal has an undeterminable category and is consequently unclear (Art. 6 PCT).

2. It is not possible to form an opinion on the novelty, inventiveness and industrial applicability of the subject-matter of the claims until a set of claims is filed clearly relating to a single invention, including a reasonable number of independent claims which define all the essential features of the invention (cf. PCT Guidelines, Chapter III, 4.4).
3. A prerequisite in order to be able to form an opinion on novelty, inventive step and industrial applicability (see Article 33 (1) PCT) is that the subject-matter claimed is clear (Art. 6 PCT).

This means in particular that **the subject-matter claimed**

- is comprehensible in itself,
- is related to a technical apparatus or method,
- is clear with respect to its category,
- includes all essential features,
- avoids definitions referring to results to be achieved and
- is clear with respect to its terminology used (consistency of terminology).

Further prerequisites defined in Article 6 PCT are support by the description and well defined scope of protection of the subject-matter claimed.

In the present case, independent Claims 1, 5, 9 to 14 do not meet the above-mentioned requirements in several aspects and to such an extent that no meaningful opinion could be formed. This is because the present wording is so vague and can be interpreted in so many ways that the matter for which protection is sought cannot be determined.

(Further reasons are given in sections IV and VIII.)

With respect to SECTION IV:

1. The Applicant was advised that all independent claims must be linked by a single inventive concept (Rule 13.1 PCT).

In the present case however, this requirement is not met. There are **two** separate inventions or groups of inventions, as follows:

- I. Claims 1 to 12 claim devices, systems or methods in which each successive packet message is associated with a priority label. The essential part of the independent claims is that each priority label appears in a predetermined cyclic sequence and the *positions* of the labels in the cyclic sequence being such as to maximise, for each label in the sequence, that the number of consecutive lower priority labels between that label and the nearest label of equal or higher priority. Hence all packet messages are considered, i.e. **none of the packets is dropped**, and all packet messages are labelled according to a particular scheme.
- II. Independent Claims 13 and 14 claim a Link Manager or a method of managing the sending of packet messages from a plurality of sources on a link respectively. According to these claims, packet messages from a sequence are successively **dropped** until the remaining packet messages of the sequence are as evenly spaced as possible. Hence, only some packet messages are considered and packet dropping is used as an organizational measure.

These separate inventions / groups of invention are not so linked as to form a single general inventive concept.

The method steps / apparatus features applied in inventions I or II as well as the problems to be solved by the different groups of inventions are quite different from each other. Therefore, no single general inventive concept can be ascribed to the inventions specified above.

The groups of inventions are neither linked by a single general inventive concept, nor do they fulfil the requirement of Rule 13.2 PCT that an international patent application may include a group of inventions if there is a technical relationship among those inventions involving **one or more of the same, or corresponding**

**INTERNATIONAL PRELIMINARY
EXAMINATION REPORT - SEPARATE SHEET**

International application No. PCT/GB00/01135

special technical features which make as a whole a contribution to the state of the art.

2. The Applicant has argued that the subject-matter of Claims 13 to 14 would relate to devices **complementary** to those claimed in claims 1 to 4. However "complementary" means different and thus non-unified under a single general inventive concept with the *same* special technical inventive features. Hence, it was not understood why the requirements of Rule 13.1 and 13.2 PCT should be met.

With respect to SECTION VII:

- 1). The independent claims have not been properly cast in the two-part form, with those features which in combination are part of the prior art being placed in the preamble. Thus they do not meet the requirements of Rule 6.3 (b) PCT.
- 2). Reference signs in parentheses have not been inserted neither to the preamble nor to the characterising portion of the claims to increase their intelligibility. Hence, the criteria of Rule 6.2 (b) PCT are not met.
- 3). None of the documents **D1** to **D3** mentioned in the international search report have been identified in the description nor has the relevant background art disclosed therein been briefly discussed. Thus, the requirements of Rule 5.1 (a) (ii) PCT are not fulfilled.

With respect to SECTION VIII:

1. The independent Claims 1, 5, 9, 10, 11, 12, 13 and 14, 15 have overlapping scopes. The various definitions of the invention given in these independent claims are such that the claims as a whole are not clear and concise, contrary to Article 6 PCT. The claims do not include only the minimum necessary number of independent claims in any one category, with dependent claims as appropriate (Rule 6.4 (a)-(c) PCT).

2. The subject-matter of Claim 1 does not comply with the requirements of Art. 6 PCT with respect to clarity.
- 2a. The definition "said priority labels each representing one of a plurality of priority levels and the positions of the labels in the cyclic sequence being such as to maximise, for each label in the sequence, the number of consecutive lower priority labels between that label and the nearest label in the sequence of equal or higher priority" is unclear and cannot be understood.
A similar objection is raised for Claims 9, 10 and 11.
- 2b. Independent Claim 1 does not meet the requirements of Article 6 PCT in that the matter for which protection is sought is not clearly defined. The claim attempts to define the subject-matter **in terms of the result to be achieved** ("the positions of the labels in the cyclic sequence being such as to maximise, for each label in the sequence, the number of consecutive lower priority labels between that label and the nearest label in the sequence of equal or higher priority"). In this instance, however, such a formulation is not tolerable because it seems possible to define the subject-matter in more concrete terms, viz. in terms how the effect is to be achieved.
This means in other words that the skilled person cannot find clear instructions in the claim how said means, arranged to associate a priority label with each successive packet message, is exactly able to find or choose a priority label for each respective packet message, i.e which algorithm is to be used. Consequently, the scope of protection for Claim 1 cannot be determined. Since the subject-matter of Claim 1 specifies no precise instructions with respect to the association process of a priority label, a comparison of this imprecise subject-matter including a plurality of conceivable realizations for the maximisation of consecutive lower priority labels with the state of the art cited is impossible. Therefore, no examination on novelty, inventive step and industrial applicability of the subject-matter claimed is possible.

Yet with the help of the description (cf. page 7, lines 10, to page 8, line 2; page 12, lines 24 to 33; page 14, lines 1 to 13; page 21, lines 5 to 10) no clarification is possible. The cited part of the description shows a structure as illustrated in reference 1 (see annex). When considering message 3 having priority 2 or label 2

and using the definition of Claim 1, i.e. the positions of the labels in the cyclic sequence (0-7-2-4-1-6-3-5) being such as to *maximise*, for each label (e.g. 2) in the sequence, the **number** of consecutive lower priority labels (i.e. 0 and 1) **between** that label (2) and the nearest label (7) in the sequence of equal or higher priority, it is apparent from reference 1 that the number of consecutive lower priority labels between the label 2 and the nearest higher priority label 7 is zero. Hence, the number of lower priority labels is not maximised although it should tend towards two. Additionally, the description merely discloses a particular cyclic sequence (0-7-2-4-1-6-3-5) and neither discloses another cyclic sequence nor discloses a universal mathematical algorithm which fulfills the result definition in Claim 1. Hence, it seems that the subject-matter claimed is insufficiently disclosed by the application (Art. 5 PCT) except the clarity objections raised above (Art. 6 PCT).

A similar objection is valid for Claims 7, 9, 10 and 11.

- 2c. Thus, Claims 1, 7, 9, 10 and 11 are unclear, do not define the matter for which protection is sought and therefore do at least not fulfil the requirements of Art. 6 PCT.
- 3a. Independent Claim 13 does not meet the requirements of Article 6 PCT in that the matter for which protection is sought is not clearly defined. The claim attempts to define the subject-matter **in terms of the result to be achieved** ("... such that successively dropping packet messages from each sequence on a priority basis leaves the remaining packet messages of the sequence *as evenly spaced with respect to the original sequence as possible*"). In this instance, however, such a formulation is not tolerable because it seems possible to define the subject-matter in more concrete terms, viz. in terms how the ~~effect~~ effect is to be achieved.
- 3b. The specification in Claim 13 "successively dropping packet messages from the sequence on a priority basis leaves the remaining packet messages of the sequence *as evenly spaced with respect to the original sequence as possible*" is unclear with respect to the part in italics. Hence, the scope of protection cannot be determined.

Yet with the help of the description (cf. page 7, lines 10, to page 8, line 2; page 12, lines 24 to 33; page 14, lines 1 to 13; page 21, lines 5 to 10) no clarification is possible. The cited part of the description shows a structure as illustrated in reference 2 (see annex). An evenly spaced message-packet-distribution with respect to the original sequence is given in the examples when

- a packet with prio 0 has been dropped,
- packets with prios 0 and 1 have been dropped, and
- packets with prios 0 to 3 have been dropped.

In all other cases, the remaining packets of the sequence are not as evenly spaced *with respect to the original sequence as possible* (note: the description on page 12, lines 31 to 33, teaches that "the remaining packets were still equally spaced in time and ...").

With respect to reference 2, illustration case when prios 0 to 2 have been dropped, the definition in Claim 13 "as evenly spaced with respect to the original sequence as possible" would mean that the five messages would occupy e.g. positions 1, 3, 5, 7, 8 (instead of 2, 4, 6, 7 and 8).

With respect to reference 2, illustration case when prios 0 to 4 have been dropped, the definition in Claim 13 "as evenly spaced with respect to the original sequence as possible" would mean that the three messages would occupy e.g. positions 1, 4 and 7 (instead of 2, 6 and 8).

With respect to reference 2, illustration case when prios 0 to 5 have been dropped, the definition in Claim 13 "as evenly spaced with respect to the original sequence as possible" would mean that the two messages would occupy e.g. positions 3 and 6 (instead of 2 and 6).

With respect to reference 2, illustration case when prios 0 to 6 have been dropped, the definition in Claim 13 "as evenly spaced with respect to the original sequence as possible" would mean that the last message would occupy e.g. position 4 or 5 (instead of 2).

Hence, it seems that the subject-matter claimed is insufficiently disclosed by the application (Art. 5 PCT) except the clarity objections raised (Art. 6 PCT).

- 3c. The addition "as possible" in the respective independent claims causes clarity problems. Hence this specification renders the respective claims unclear (Art. 6 PCT). Moreover, this addition is regarded as a vague and equivocal expression

**INTERNATIONAL PRELIMINARY
EXAMINATION REPORT - SEPARATE SHEET**

International application No. PCT/GB00/01135

which is to be avoided according to the PCT-Guidelines (cf. PCT-Gazette, Section IV, III-4.5).

In this connection, it should be noted that the packet messages are indefinite with respect to their lengths. The packet messages could have a variable length or a fixed length. In the case when the packet messages are of variable length, it is very difficult to achieve an even spacing.

- 3d. Claim 14 specifies a source. This source sends "sequences" (cf. page 35, line 8). In line 10 and 11 on the same page, reference is made to "the sequence". It is unclear which particular sequence could be meant since several sequences are defined and a single sequence has no antecedent.
- 3e. The subject-matter of independent Claims 13 and 14 is thus so unclear that the subject-matter claimed is excluded from an examination with respect to Article 33 (1) PCT.

References 1 and 2 are to follow (two further pages)

PCT

INTERNATIONAL SEARCH REPORT

(PCT Article 18 and Rules 43 and 44)

Applicant's or agent's file reference A25771 WO	FOR FURTHER ACTION see Notification of Transmittal of International Search Report (Form PCT/ISA/220) as well as, where applicable, item 5 below.	
International application No. PCT/GB 00/ 01135	International filing date (day/month/year) 24/03/2000	(Earliest) Priority Date (day/month/year) 31/03/1999
Applicant BRITISH TELECOMMUNICATIONS PUBLIC LIMITED COMPANY		

This International Search Report has been prepared by this International Searching Authority and is transmitted to the applicant according to Article 18. A copy is being transmitted to the International Bureau.

This International Search Report consists of a total of 3 sheets.

☒ It is also accompanied by a copy of each prior art document cited in this report.

1. Basis of the report

a. With regard to the **language**, the international search was carried out on the basis of the international application in the language in which it was filed, unless otherwise indicated under this item.

☐ the international search was carried out on the basis of a translation of the international application furnished to this Authority (Rule 23.1(b)).

b. With regard to any **nucleotide and/or amino acid sequence** disclosed in the international application, the international search was carried out on the basis of the sequence listing :

☐ contained in the international application in written form.

☐ filed together with the international application in computer readable form.

☐ furnished subsequently to this Authority in written form.

☐ furnished subsequently to this Authority in computer readable form.

☐ the statement that the subsequently furnished written sequence listing does not go beyond the disclosure in the international application as filed has been furnished.

☐ the statement that the information recorded in computer readable form is identical to the written sequence listing has been furnished

2. ☐ **Certain claims were found unsearchable** (See Box I).

3. ☐ **Unity of Invention is lacking** (see Box II).

4. With regard to the **title**,

☒ the text is approved as submitted by the applicant.

☐ the text has been established by this Authority to read as follows:

5. With regard to the **abstract**,

☒ the text is approved as submitted by the applicant.

☐ the text has been established, according to Rule 38.2(b), by this Authority as it appears in Box III. The applicant may, within one month from the date of mailing of this international search report, submit comments to this Authority.

6. The figure of the **drawings** to be published with the abstract is Figure No.

☒ as suggested by the applicant.

☐ because the applicant failed to suggest a figure.

☐ because this figure better characterizes the invention.

4A

☐ None of the figures.

REPLACED BY
ART 34 AMDT

they all were a short time ago) to within a relatively small error rather than knowing where a subset of the creatures are to within a very small error (where they are now) with the remainder of the (highly mobile hostile) creature's positions only being known with a relatively large error (where they were a while ago).

5

It will be clear that a simple hierarchically oriented scheme such as a layered video coding scheme cannot provide for such a situation. As indicated above, in such a scheme all available information may be required from the most important source entities at the expense of receiving little, if any, information from the less important source entities whereas having regard to the problem under consideration, at least some information is required from all the source entities, each being nominally of the same importance.

Thus, according to one aspect of the present invention, there is provided a packet message source comprising: means arranged to include a respective packet message payload in each packet message of a sequence of packet messages; means arranged to associate a priority label with each successive packet message in said sequence in accordance with a predetermined cyclic sequence of such labels; said priority labels each representing one of a plurality of priority levels and the position of each label in the cyclic sequence being such that the number of consecutive lower priority labels between that label and the nearest label of equal or higher priority is substantially maximised; and means arranged to send such packet messages.

Advantageously, this will ensure that, when a number of such packet message sources are sending such packet messages and must contend for limited bandwidth such that packets are dropped as necessary in accordance with their priority labels, as far as possible packet messages representing regular updates from all the sources can be sent to a receiving client, rather than all the updates from a subset of the sources and few, if any, updates from the remaining sources as with the prior art.

30

A packet message system, a method of packet messaging and a method of operating a packet messaging system are also provided.

portion 304. It is to be noted that not all the fields in the packet header 302 will be simultaneously required for all the embodiments of the invention.

5 A first field 306 in the packet header 302 contains an indication of the identity of the source entity. A second field 308 in the packet header 302 contains an indication of the identity of the intended destination entity. With only one destination in this first embodiment, this field need not be set. A third field 310 in the packet header 302 contains an indication of assigned priority, on a given priority scale.

10 According to the invention, the third field 310, identifying a priority setting, will be set as follows. Each individual packet message is assigned a priority label, each priority label representing one of a plurality of priority levels. Successive packets, considered in the sequence in which they were generated by each packet message source, are allocated one of these priority labels in accordance with a predetermined
15 cyclic label sequence. Crucially, the position of each label in this cyclic sequence is such that the number of consecutive lower priority labels between it and the nearest label of equal or higher priority is substantially maximised.

The cyclic priorities thus vary such that if packets were discarded on a priority basis,
20 the remaining packets in the notional sequence would be left as evenly spaced as possible. It will be noted that sequence lengths of powers of two are advantageous in this respect.

For a simple example with a priority scale of 0 (lowest priority) to n (highest priority)
25 and where $n=7$, a corresponding suitable cyclic variation in priority is a sequence of eight messages (equally spaced in time) with priorities set at 0, 7, 2, 4, 1, 6, 3 and 5 respectively. The 0, 7, 2, 4, 1, 6, 3, 5 sequence will ensure that as successive low priorities have to be dropped the remaining packets will be as evenly spaced as possible. By way of example, if half the packets had to be dropped from such a
30 sequence of eight messages due to a traffic load of twice link capacity (thus becoming [], 7, 2, 4, 1, 6, 3, 5 then [], 7, 2, 4, [], 6, 3, 5 then [], 7, [], 4, [], 6, 3, 5, and finally [], 7, [], 4, [], 6, [], 5) it will be seen that the remaining packets are

The third field 310, identifying a priority setting, will be set as with the first embodiment. Each individual packet message is assigned a priority setting on a scale of for example, 0 (lowest priority) to n (highest priority). Again crucially, a sequence of $n + 1$ such messages is sent out from the source entity, not with a fixed priority
5 assigned to the source entity in question, but with cyclically varying priorities. The cyclic priorities vary such that if packets were discarded on a priority basis, the remaining packets in the sequence would be left as evenly spaced as possible.

Again, for a simple example where $n = 7$, a corresponding suitable cyclic variation in
10 priority is a sequence of eight messages with priorities set at 0, 7, 2, 4, 1, 6, 3 and 5 respectively. As indicated above, the 0, 7, 2, 4, 1, 6, 3, 5 sequence will ensure that as successive low priority packet messages have to be dropped, the remaining packet messages will be as evenly spaced as possible.

15 In this second embodiment the fourth field 312 will be set with a time-to-live period. It will be appreciated that rather than all source entities needing to have the same packet message sending periodicity, varying time-to-live indications allow each source entity to set their own periodicity. For example, as a result of its nature, a given source entity might only need to send updates every 20 nominal time periods, in
20 which case a time-to-live period could be set at 20 nominal time periods. After 20 such nominal time periods, the given source entity will send out a new update so the older update may be discarded.

In this second embodiment, the fifth field 314 in the packet header 302, giving an
25 indication of the packet payload type, is not set.

Streams of such packet messages, bearing cyclic priority settings, as discussed above, are sent out from the source entities 100, 100'. These packet messages are then passed, via the respective first and second server computer output ports 106,
30 106' and first and second links 108, 108', to the link management computer 110.

Having regard to Figure 5, in a first step 500, after these packet messages are received at the first and second link management computer input ports 112, 112', they are passed to the Link Manager process 114.

- 5 The Link Manager process 114 will again maintain a simple packet message queue 116, ordered on priority, as with the first embodiment.

The Link Manager process 114 is provided with a clock (not shown) keeping a local time measured cyclically in, for example milliseconds. In a second step 502, the Link
10 Manager process 114 reads the time-to-live setting in the fourth field 312 of the respective packet headers 302 and in a third step 504, adds the local clock time to the time-to-live setting assigned by the source entity, to create an adjusted time-to-live. In a fourth step 506, this adjusted time-to-live is then written back into the fourth field 312 of the packet header 302.

15

It will be appreciated that, by way of an alternative, the packet message and an associated arrival time could instead be stored together in the queue 116. This would have the advantage of not writing over the source entity set time-to-live but would have the disadvantage of a commensurately larger queue memory requirement.

20

It will be appreciated that the only significant delay which may be experienced by a given packet message will be that resulting from being held in a queue. If, by way of the example used above, a given packet message is received by the Link Manager process 114 with a time-to-live indication of 20 nominal units and the Link Manager
25 process local time was $t = 100$ nominal units, then the adjusted time-to-live written into the fourth field of the packet header would be 120 nominal units. If, for reasons discussed below, this packet message was queued for more than the 20 nominal units lifetime, by which time the Link Manager process local time would be $t = 120$ nominal units, then the packet message should be timed-out. The source entity
30 should have sent, and the Link Manager process 116 should have received, another packet message by this time.

In a fifth step 506, the Link manager process 114 also reads each priority setting in the third field 310 of the respective packet headers 302 and then, in a sixth step 510, sorts the packet messages into the simple queue 116 ordered on the basis of the priority setting. Again, the packet message might, for example, be sorted into the queue 116 at the rear of the group of packet messages with the same priority setting. As indicated above, again, by way of an alternative, the portions of the simple queue 116 with the same priority settings could be queued separately.

The process illustrated in Figure 6 is to be understood as being executed in parallel with that illustrated in Figure 5.

Having regard to Figure 6, in a first step 600, as each new packet message moves up to the head of the queue 116 for sending onto the link 120, the Link Manager process 114 will again read the adjusted time-to-live setting from the fourth field 312 of the packet header 302 of the packet message at the head of the queue 116.

In a second step 602, the Link Manager 114 will compare this adjusted time-to-live with the Link Manager process local time. In a third step 604, if the Link Manager process local time has passed the adjusted time-to-live of the packet message, then that packet message must be considered as being timed-out and will be discarded by the Link Manager process 114. If, however, the Link Manager process local time has not passed the adjusted time-to-live of the packet message, then the packet message will not have been queued for longer than its assigned time-to-live and in a fourth step 606, will next be sent, via the output port of the link management computer 118, onto the link and to the receiving entity at the client 120. The Link Manager process 114 will again test the link 120 for capacity to send the packet message at the head of the priority queue 116 in doing so, as with the first embodiment.

It will be appreciated that for optimal utilisation of the bandwidth available on the low-bandwidth link, a step of header compression (not shown) may take place. Portions of the header 302 no longer necessary since the packet message is to be sent out over the low-bandwidth link 120, for example, the priority setting, may be stripped off.

Link Manager process 114 might then have sent the first message over the low bandwidth link to the receiving entity host computer.

Had the Link Manager process 114 done so, it would have had the result that the latest positional update message received by the client would in fact have dated from one and two time periods respectively before the positional updates carried by the second and third messages which had already been received by the client. Instead, the time-to-live setting on the message will cause the message to be discarded from the Link Manager process queue 116 before it can be sent over the low bandwidth link 120.

It will be appreciated, as indicated above, that the longest delay, indeed the only significant delay, the packet message will experience before transmission, is that resulting from being queued. The time-to-live setting, apart from its advantage in the prevention of an erroneous re-ordering of packet messages as explained above, is also of more general assistance in queue memory management. The packet message queue 116 could be swept, for example, when the queue 116 reaches a certain pre-determined length. Memory space otherwise taken up by timed-out packet messages may thereby be freed up. It would still be necessary, however, to check for timing-out on the sending of each packet. Alternatively, the queue 116 could be swept before it had reached this certain length. This would free up memory space otherwise taken by timed-out packets that might otherwise have lived on in a queue 116 shorter than the certain length.

The fruits of this approach to management of the low bandwidth link by the Link Manager process, combining cyclic priority settings with time-to-live settings are well demonstrated in and will be discussed having regard to, Figure 7 and Figure 8.

A simulation was created as follows. A 'Server' process was set up on a single PC, with a small number of initial entities, each of which possess position state. A 'Client' process was set up with an identical state on the same machine and the two were connected via a 28.8 Kbit/s UDP/IP link via a 'reflector' machine (simply bouncing packets from the server back to the client).

CLAIMS

1. A packet message source comprising:
 - 5 means arranged to include a respective packet message payload in each packet message of a sequence of packet messages;

means arranged to associate a priority label with each successive packet message in said sequence in accordance with a predetermined cyclic sequence of such labels;
 - 10 said priority labels each representing one of a plurality of priority levels and the position of each label in the cyclic sequence being such that the number of consecutive lower priority labels between that label and the nearest label of equal or higher priority is substantially maximised; and
 - 15 means arranged to send such packet messages.
2. A packet message source as claimed in claim 1, wherein said packet message source has an associated dynamic state and said respective packet message
20 payload comprises a source state update message.
3. A packet message source as claimed in claim 1 or claim 2 further comprising:
 - 25 means arranged to associate a time-to-live setting with each packet message.
4. A packet message source as claimed any one of claims 1 to 3 further comprising:
30 means arranged to associate a packet message source identity with each packet message; and

means arranged to associate a packet message payload type setting with each packet message.

5 5. A packet messaging system comprising:

a plurality of packet message sources according to any one of claims 1 to 4 ;and

a communications link interface;

10

said communications link interface comprising:

an input port for receiving packet messages from said plurality of packet message sources;

15

means arranged to read a priority label associated with each received packet message;

a queue for queuing received packet messages in order of their associated priority labels; and

20

an output port for sending each packet message at the head of said queue onto a communications link.

25 6. A packet messaging system according to claim 5 when depending on claim 3, said communications link interface further comprising:

means arranged to read a packet message time-to-live setting associated with each received packet message;

30

means arranged to associate with each respective packet message an indication of the period of time that packet message has been queued ;and

means arranged to discard each packet message whose associated indication indicates that that packet message has been queued for a period of time longer than the associated packet message time-to-live setting.

- 5 7. A packet messaging system according to claim 5 when depending on claim 3, said communications link interface further comprising:

a clock; and

- 10 means arranged to replace the packet message time-to-live setting associated with each packet message with an adjusted time-to-live setting resulting from the addition of the time-to-live setting read from the packet message and the local time at which the packet was received;

- 15 and in which the means arranged to associate an indication of a period of time with each packet message, associates with at least one packet message, the result of said adjusted time-to-live read from each said packet message minus the local time at which said packet message arrived such that the means arranged to discard each packet message are so arranged to discard a packet message if said associated sum
20 is less than or equal to zero.

8. A packet messaging system according to claim 5 when depending on claim 4, said communications link interface further comprising:

- 25 means arranged to read a packet message source identity from each received packet message;

means arranged to read a packet message payload type setting associated with each received packet message;

30

means arranged to test the packet message queue for a queued packet message with an associated source identity matching that of the received packet message;

means arranged to read the priority label associated with the matching queued packet message;

means arranged to sort into the queue on the basis of priority label a packet message
5 replacing the matched received and queued packet messages, having the associated source identity of the matched received and queued packet messages, the payload of the received packet message and whichever of the associated priority label of the matched received and queued packet message represents the higher priority.

10 9. A packet messaging system comprising:

a plurality of packet message sources each comprising:

means arranged to include a respective packet message payload in each packet
15 message of a sequence of packet messages; and

means arranged to send such packet messages; and

a communications link interface comprising:

20

an input port for receiving packet messages from said plurality of packet message sources;

a queue for queuing packet messages in order of respective allocated priority labels;
25 successive packets, considered in said sequence at each one of said plurality of packet message sources, having been allocated said priority labels in accordance with a predetermined cyclic sequence of such labels; said priority labels each representing one of a plurality of priority levels and the position of each label in the cyclic sequence being such that the number of consecutive lower priority labels between it
30 and the nearest label of equal or higher priority is substantially maximised; and

an output port for sending each packet message at the head of said queue onto a communications link.

10. A method of packet messaging comprising:

including a respective packet message payload in each packet message of a sequence
5 of packet messages;

associating a priority label with each successive packet message of said sequence in
accordance with a predetermined cyclic sequence of such labels; said priority labels
each representing one of a plurality of priority levels and the position of each label in
10 the cyclic sequence being such that the number of consecutive lower priority labels
between said label and the nearest label of equal or higher priority is substantially
maximised; and

sending such packet messages.

15

11. A method of operating a packet messaging system including a plurality of
packet message sources and a communications link interface, said method
comprising:

20 allocating successive packets, considered in their original sequence at each one of the
packet message sources, priority labels in accordance with a predetermined cyclic
sequence of such labels; said priority labels each representing one of a plurality of
priority levels and the position of each label in the cyclic sequence being such that
the number of consecutive lower priority labels between it and the nearest label of
25 equal or higher priority is substantially maximised;

queuing said packet messages in a queue in order of respective allocated priority
labels at said communications link interface; and

30 sending each packet message at the head of said queue onto a communications link.

12. A computer readable storage medium, said medium embodying computer readable code for reading into a computer and executable by said computer to perform the method of claim 10 or claim 11.

5 13. A Link Manager for managing the sending of packet messages on a link;

the packet messages originating from a plurality of packet message sources; each packet message source having an associated state; each packet message including an associated packet message priority setting on a scale of n to m
10 and a packet message source state update;
each packet message having been sent by a source sending sequences of $m - n + 1$ packet messages cycling through each priority setting of said priority setting scale such that successively dropping packet messages from the sequence on a priority basis leaves the remaining packet
15 messages of the sequence as evenly spaced as possible;

the Link Manager comprising:

at least one packet message input port to receive such packet messages;
20 means arranged to read the priority setting associated with each such received packet message

means arranged to sort each such received packet message into a packet message
25 queue on the basis of the priority setting associated with each such received packet message;

means arranged to test the link for sufficient capacity to send the packet message at the head of the queue; and

30 means arranged to send the packet message at the head of the queue, when the link has sufficient capacity, out through at least one output port onto the link.

14. A method of managing the sending of packet messages from a plurality of sources on a link;

5 the packet messages originating from a plurality of packet message sources;
each packet message source having an associated state; each packet message
including an associated packet message priority setting on a scale of n to m
and a packet message source state update;
each packet message having been sent by a source sending sequences of $m -$
10 $n + 1$ packet messages cycling through each priority setting of said priority
setting scale such that successively dropping packet messages from the
sequence on a priority basis leaves the remaining packet
messages of the sequence as evenly spaced as possible;

the method comprising:

15 receiving such packet messages at at least one input port of a link manager;

reading the priority setting associated with each such received packet message;

20 sorting each such received packet message into a packet message queue on the
basis of the priority setting associated with each such received packet message;
and

sending the packet message at the head of the queue out through at least one
25 output port of the link manager onto the link.

message1	message2	message3	message4	message5	message6	message7	message8
prio0	prio7	prio2	prio4	prio1	prio6	prio3	prio5

>t

eight packet messages, priorities from 0 (low) to 7 (high)

reference 2

position

- ① ② ③ ④ ⑤ ⑥ ⑦ ⑧

message2	message3	message4	message5	message6	message7	message8
prio7	prio2	prio4	prio1	prio6	prio3	prio5

>t

prio 0 has been dropped

message2	message3	message4	message6	message7	message8
prio7	prio2	prio4	prio6	prio3	prio5

>t

prios 0 and 1 have been dropped

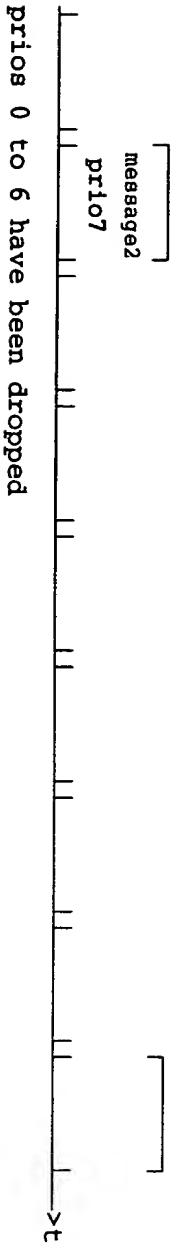
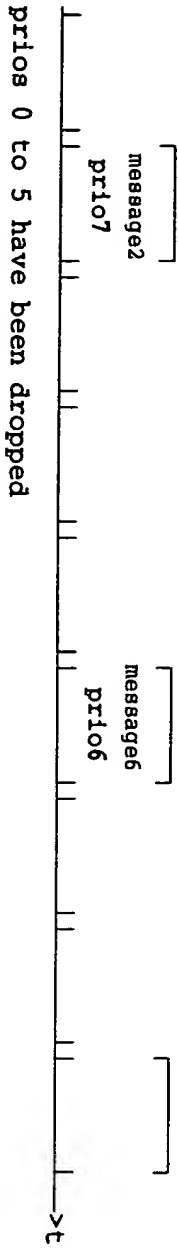
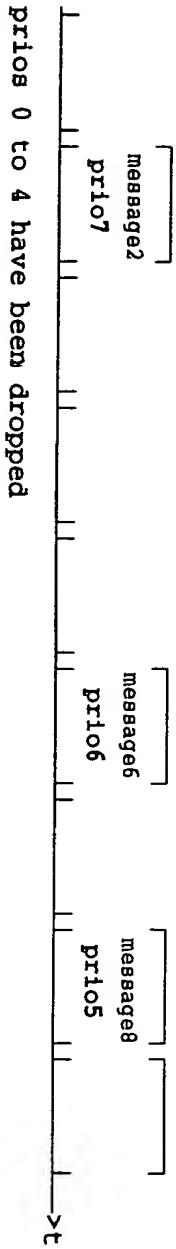
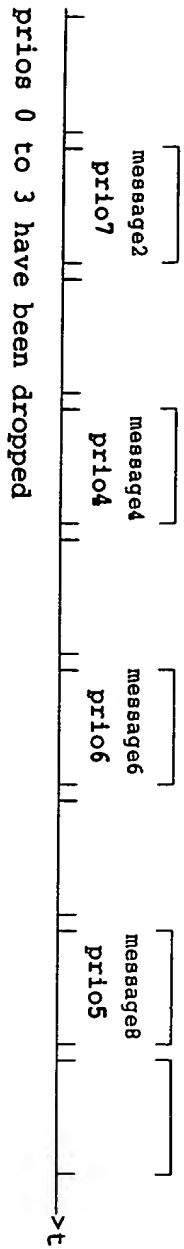
message2	message4	message6	message7	message8
prio7	prio4	prio6	prio3	prio5

>t

prios 0 to 2 have been dropped

position

- ①
- ②
- ③
- ④
- ⑤
- ⑥
- ⑦
- ⑧



they all were a short time ago) to within a relatively small error rather than knowing where a subset of the creatures are to within a very small error (where they are now) with the remainder of the (highly mobile hostile) creature's positions only being known with a relatively large error (where they were a while ago).

5

It will be clear that a simple hierarchically oriented scheme such as a layered video coding scheme cannot provide for such a situation. As indicated above, in such a scheme all available information may be required from the most important source entities at the expense of receiving little, if any, information from the less important source entities whereas having regard to the problem under consideration, at least some information is required from all the source entities, each being nominally of the same importance.

Thus, according to one aspect of the present invention, there is provided a packet message source comprising: means arranged to include a respective packet message payload in each packet message of a sequence of packet messages; means arranged to associate a priority label with each successive packet message in said sequence in accordance with a predetermined cyclic sequence of such labels; said priority labels each representing one of a plurality of priority levels and the positions of the labels in the cyclic sequence being such as to maximise, for each label in the sequence, the number of consecutive lower priority labels between that label and the nearest label in the sequence of equal or higher priority; and means arranged to send such packet messages.

Advantageously, this will ensure that, when a number of such packet message sources are sending such packet messages and must contend for limited bandwidth such that packets are dropped as necessary in accordance with their priority labels, as far as possible packet messages representing regular updates from all the sources can be sent to a receiving client, rather than all the updates from a subset of the sources and few, if any, updates from the remaining sources as with the prior art.

A packet message system, a method of packet messaging and a method of operating a packet messaging system are also provided.

portion 304. It is to be noted that not all the fields in the packet header 302 will be simultaneously required for all the embodiments of the invention.

A first field 306 in the packet header 302 contains an indication of the identity of the source entity. A second field 308 in the packet header 302 contains an indication of the identity of the intended destination entity. With only one destination in this first embodiment, this field need not be set. A third field 310 in the packet header 302 contains an indication of assigned priority, on a given priority scale.

- 10 According to the invention, the third field 310, identifying a priority setting, will be set as follows. Each individual packet message is assigned a priority label, each priority label representing one of a plurality of priority levels. Successive packets, considered in the sequence in which they were generated by each packet message source, are allocated one of these priority labels in accordance with a predetermined
- 15 cyclic label sequence. Crucially, the positions of the labels in this cyclic sequence are such as to maximise, for each label in the sequence, the number of consecutive lower priority labels between it and the nearest label in the sequence of equal or higher priority.
- 20 The cyclic priorities thus vary such that if packets were discarded on a priority basis, the remaining packets in the notional sequence would be left as evenly spaced with respect to the original sequence as possible. It will be noted that sequence lengths of powers of two are advantageous in this respect.
- 25 For a simple example with a priority scale of 0 (lowest priority) to n (highest priority) and where $n = 7$, a corresponding suitable cyclic variation in priority is a sequence of eight messages (equally spaced in time) with priorities set at 0, 7, 2, 4, 1, 6, 3 and 5 respectively. The 0, 7, 2, 4, 1, 6, 3, 5 sequence will ensure that as successive low priorities have to be dropped the remaining packets will be as evenly spaced as
- 30 possible. By way of example, if half the packets had to be dropped from such a sequence of eight messages due to a traffic load of twice link capacity (thus becoming [], 7, 2, 4, 1, 6, 3, 5 then [], 7, 2, 4, [], 6, 3, 5 then [], 7, [], 4, [], 6, 3, 5, and finally [], 7, [], 4, [], 6, [], 5) it will be seen that the remaining packets are

The third field 310, identifying a priority setting, will be set as with the first embodiment. Each individual packet message is assigned a priority setting on a scale of for example, 0 (lowest priority) to n (highest priority). Again crucially, a sequence of $n + 1$ such messages is sent out from the source entity, not with a fixed priority
5 assigned to the source entity in question, but with cyclically varying priorities. The cyclic priorities vary such that if packets were discarded on a priority basis, the remaining packets in the sequence would be left as evenly spaced with respect to the original sequence as possible.

10 Again, for a simple example where $n = 7$, a corresponding suitable cyclic variation in priority is a sequence of eight messages with priorities set at 0, 7, 2, 4, 1, 6, 3 and 5 respectively. As indicated above, the 0, 7, 2, 4, 1, 6, 3, 5 sequence will ensure that as successive low priority packet messages have to be dropped, the remaining packet messages will be as evenly spaced with respect to the original sequence as possible.

15

In this second embodiment the fourth field 312 will be set with a time-to-live period. It will be appreciated that rather than all source entities needing to have the same packet message sending periodicity, varying time-to-live indications allow each source entity to set their own periodicity. For example, as a result of its nature, a given
20 source entity might only need to send updates every 20 nominal time periods, in which case a time-to-live period could be set at 20 nominal time periods. After 20 such nominal time periods, the given source entity will send out a new update so the older update may be discarded.

25 In this second embodiment, the fifth field 314 in the packet header 302, giving an indication of the packet payload type, is not set.

Streams of such packet messages, bearing cyclic priority settings, as discussed above, are sent out from the source entities 100, 100'. These packet messages are
30 then passed, via the respective first and second server computer output ports 106, 106' and first and second links 108, 108', to the link management computer 110.

Having regard to Figure 5, in a first step 500, after these packet messages are received at the first and second link management computer input ports 112, 112', they are passed to the Link Manager process 114.

- 5 The Link Manager process 114 will again maintain a simple packet message queue 116, ordered on priority, as with the first embodiment.

The Link Manager process 114 is provided with a clock (not shown) keeping a local time measured cyclically in, for example milliseconds. In a second step 502, the Link
10 Manager process 114 reads the time-to-live label in the fourth field 312 of the respective packet headers 302 and in a third step 504, adds the local clock time to the time-to-live label assigned by the source entity, to create an adjusted time-to-live. In a fourth step 506, this adjusted time-to-live is then written back into the fourth field 312 of the packet header 302.

15

It will be appreciated that, by way of an alternative, the packet message and an associated arrival time could instead be stored together in the queue 116. This would have the advantage of not writing over the source entity set time-to-live but would have the disadvantage of a commensurately larger queue memory requirement.

20

It will be appreciated that the only significant delay which may be experienced by a given packet message will be that resulting from being held in a queue. If, by way of the example used above, a given packet message is received by the Link Manager process 114 with a time-to-live indication of 20 nominal units and the Link Manager
25 process local time was $t = 100$ nominal units, then the adjusted time-to-live written into the fourth field of the packet header would be 120 nominal units. If, for reasons discussed below, this packet message was queued for more than the 20 nominal units lifetime, by which time the Link Manager process local time would be $t = 120$ nominal units, then the packet message should be timed-out. The source entity
30 should have sent, and the Link Manager process 116 should have received, another packet message by this time.

In a fifth step 506, the Link manager process 114 also reads each priority setting in the third field 310 of the respective packet headers 302 and then, in a sixth step 510, sorts the packet messages into the simple queue 116 ordered on the basis of the priority setting. Again, the packet message might, for example, be sorted into the queue 116 at the rear of the group of packet messages with the same priority setting. As indicated above, again, by way of an alternative, the portions of the simple queue 116 with the same priority settings could be queued separately.

The process illustrated in Figure 6 is to be understood as being executed in parallel with that illustrated in Figure 5.

Having regard to Figure 6, in a first step 600, as each new packet message moves up to the head of the queue 116 for sending onto the link 120, the Link Manager process 114 will again read the adjusted time-to-live label from the fourth field 312 of the packet header 302 of the packet message at the head of the queue 116.

In a second step 602, the Link Manager 114 will compare this adjusted time-to-live with the Link Manager process local time. In a third step 604, if the Link Manager process local time has passed the adjusted time-to-live of the packet message, then that packet message must be considered as being timed-out and will be discarded by the Link Manager process 114. If, however, the Link Manager process local time has not passed the adjusted time-to-live of the packet message, then the packet message will not have been queued for longer than its assigned time-to-live and in a fourth step 606, will next be sent, via the output port of the link management computer 118, onto the link and to the receiving entity at the client 120. The Link Manager process 114 will again test the link 120 for capacity to send the packet message at the head of the priority queue 116 in doing so, as with the first embodiment.

It will be appreciated that for optimal utilisation of the bandwidth available on the low-bandwidth link, a step of header compression (not shown) may take place.

Portions of the header 302 no longer necessary since the packet message is to be sent out over the low-bandwidth link 120, for example, the priority setting, may be stripped off.

Link Manager process 114 might then have sent the first message over the low bandwidth link to the receiving entity host computer.

Had the Link Manager process 114 done so, it would have had the result that the latest positional update message received by the client would in fact have dated from one and two time periods respectively before the positional updates carried by the second and third messages which had already been received by the client. Instead, the time-to-live label on the message will cause the message to be discarded from the Link Manager process queue 116 before it can be sent over the low bandwidth link

10 120.

It will be appreciated, as indicated above, that the longest delay, indeed the only significant delay, the packet message will experience before transmission, is that resulting from being queued. The time-to-live label, apart from its advantage in the prevention of an erroneous re-ordering of packet messages as explained above, is also of more general assistance in queue memory management. The packet message queue 116 could be swept, for example, when the queue 116 reaches a certain pre-determined length. Memory space otherwise taken up by timed-out packet messages may thereby be freed up. It would still be necessary, however, to check for timing-out on the sending of each packet. Alternatively, the queue 116 could be swept before it had reached this certain length. This would free up memory space otherwise taken by timed-out packets that might otherwise have lived on in a queue 116 shorter than the certain length.

25 The fruits of this approach to management of the low bandwidth link by the Link Manager process, combining cyclic priority settings with time-to-live labels are well demonstrated in and will be discussed having regard to, Figure 7 and Figure 8.

A simulation was created as follows. A 'Server' process was set up on a single PC, with a small number of initial entities, each of which possess position state. A 'Client' process was set up with an identical state on the same machine and the two were connected via a 28.8 Kbit/s UDP/IP link via a 'reflector' machine (simply bouncing packets from the server back to the client).

30

CLAIMS

1. A packet message source comprising:

5 means arranged to include a respective packet message payload in each packet message of a sequence of packet messages;

means arranged to associate a priority label with each successive packet message in said sequence in accordance with a predetermined cyclic sequence of such labels;
10 said priority labels each representing one of a plurality of priority levels and the positions of the labels in the cyclic sequence being such as to maximise, for each label in the sequence, the number of consecutive lower priority labels between that label and the nearest label in the sequence of equal or higher priority; and

15 means arranged to send such packet messages.

2. A packet message source as claimed in claim 1, wherein said packet message source has an associated dynamic state and said respective packet message
20 payload comprises a source state update message.

3. A packet message source as claimed in claim 1 or claim 2 further comprising:

25 means arranged to associate a time-to-live label with each packet message.

4. A packet message source as claimed any one of claims 1 to 3 further comprising:

30

means arranged to associate a packet message source identity with each packet message; and

means arranged to associate a packet message payload type setting with each packet message.

5 5. A packet messaging system comprising:

a plurality of packet message sources according to any one of claims 1 to 4 ;and

a communications link interface;

10

said communications link interface comprising:

an input port for receiving packet messages from said plurality of packet message sources;

15

means arranged to read a priority label associated with each received packet message;

20

a queue for queuing received packet messages in descending order of their associated priority labels; and

an output port for sending each packet message at the head of said queue onto a communications link.

25 6. A packet messaging system according to claim 5 when depending on claim 3, said communications link interface further comprising:

means arranged to read a packet message time-to-live label associated with each received packet message;

30

means arranged to associate with each respective packet message an indication of the period of time that packet message has been queued ;and

means arranged to discard each packet message whose associated indication indicates that that packet message has been queued for a period of time longer than the associated packet message time-to-live label.

- 5 7. A packet messaging system according to claim 5 when depending on claim 3, said communications link interface further comprising:

a clock; and

- 10 means arranged to replace the packet message time-to-live label associated with each packet message with an adjusted time-to-live label resulting from the addition of the value of the time-to-live label read from the packet message and the local time at which the packet was received;

- 15 and in which the means arranged to associate an indication of a period of time with each packet message, associates with at least one packet message, the result of said adjusted time-to-live read from each said packet message minus the local time at which said packet message arrived such that the means arranged to discard each packet message are so arranged to discard a packet message if said associated sum
20 is less than or equal to zero.

8. A packet messaging system according to claim 5 when depending on claim 4, said communications link interface further comprising:

- 25 means arranged to read a packet message source identity from each received packet message;

means arranged to read a packet message payload type setting associated with each received packet message;

30

means arranged to test the packet message queue for a queued packet message with an associated source identity matching that of the received packet message;

means arranged to read the priority label associated with the matching queued packet message;

means arranged to sort into the queue on the basis of priority label a packet message
5 replacing the matched received and queued packet messages, having the associated source identity of the matched received and queued packet messages, the payload of the received packet message and whichever of the associated priority label of the matched received and queued packet message represents the higher priority.

10 9. A packet messaging system comprising:

a plurality of packet message sources each comprising:

means arranged to include a respective packet message payload in each packet
15 message of a sequence of packet messages; and

means arranged to send such packet messages; and

a communications link interface comprising:

20

an input port for receiving packet messages from said plurality of packet message sources;

means arranged to read a priority label associated with each received packet message;

25 a queue for queuing packet messages in descending order of respective allocated priority labels; successive packets, considered in said sequence at each one of said plurality of packet message sources, having been allocated said priority labels in accordance with a predetermined cyclic sequence of such labels; said priority labels each representing one of a plurality of priority levels and the positions of the labels in
30 the cyclic sequence being such as to maximise, for each label in the sequence, the number of consecutive lower priority labels between it and the nearest label in the sequence of equal or higher priority; and

an output port for sending each packet message at the head of said queue onto a communications link.

10. A method of packet messaging comprising:

5

including a respective packet message payload in each packet message of a sequence of packet messages;

associating a priority label with each successive packet message of said sequence in
10 accordance with a predetermined cyclic sequence of such labels; said priority labels each representing one of a plurality of priority levels and the positions of the labels in the cyclic sequence being such as to maximise, for each label in the sequence, the number of consecutive lower priority labels between said label and the nearest label in the sequence of equal or higher priority; and

15

sending such packet messages.

11. A method of operating a packet messaging system including a plurality of packet message sources and a communications link interface, said method
20 comprising:

allocating successive packets, considered in their original sequence at each one of the packet message sources, priority labels in accordance with a predetermined cyclic sequence of such labels; said priority labels each representing one of a plurality of
25 priority levels and the positions of the labels in the cyclic sequence being such as to maximise, for each label in the sequence, the number of consecutive lower priority labels between it and the nearest label in the sequence of equal or higher priority;

queuing said packet messages in a queue in descending order of respective allocated
30 priority labels at said communications link interface; and

sending each packet message at the head of said queue onto a communications link.

12. A computer readable storage medium, said medium embodying computer readable code for reading into a computer and executable by said computer to perform the method of claim 10 or claim 11.

5 13. A Link Manager for managing the sending of packet messages on a link;

the packet messages originating from a plurality of packet message sources; each packet message source having an associated state; each packet message including an associated packet message priority setting on a scale of n to m
10 and a packet message source state update;
each packet message having been sent by a source sending sequences of $m - n + 1$ packet messages cycling through each priority setting of said priority setting scale such that successively dropping packet messages from each sequence on a priority basis leaves the remaining packet messages of the
15 sequence as evenly spaced with respect to the original sequence as possible;

the Link Manager comprising:

at least one packet message input port to receive such packet messages;

20

means arranged to read the priority setting associated with each such received packet message

means arranged to sort each such received packet message into a packet message
25 queue on the basis of the priority setting associated with each such received packet message;

means arranged to test the link for sufficient capacity to send the packet message at the head of the queue; and

30

means arranged to send the packet message at the head of the queue, when the link has sufficient capacity, out through at least one output port onto the link.

14. A method of managing the sending of packet messages from a plurality of sources on a link;

the packet messages originating from a plurality of packet message sources;
5 each packet message source having an associated state; each packet message including an associated packet message priority setting on a scale of n to m and a packet message source state update;
each packet message having been sent by a source sending sequences of $m - n + 1$ packet messages cycling through each priority setting of said priority
10 setting scale such that successively dropping packet messages from the sequence on a priority basis leaves the remaining packet messages of the sequence as evenly spaced with respect to the original sequence as possible;

the method comprising:

15

receiving such packet messages at at least one input port of a link manager;

reading the priority setting associated with each such received packet message;

20 sorting each such received packet message into a packet message queue on the basis of the priority setting associated with each such received packet message;
and

25 sending the packet message at the head of the queue out through at least one output port of the link manager onto the link.

15. A signal comprising a sequence of packet messages, each packet message including a respective packet message payload and an associated priority label; each said priority label representing one of a plurality of priority levels, the priority labels being allocated to the successive packet message in accordance with a predetermined
30 cyclic sequence of such labels, the positions of the labels in the cyclic sequence being such as to maximise, for each label in the sequence, the number of consecutive lower priority labels between that label and the nearest label in the sequence having equal or higher priority.

16. A signal according to claim 15, wherein each packet message has an associated time-to-live label.

- 5 17. A signal according to claim 15 or claim 16, wherein each packet message includes an associated packet message source identity and an associated packet message payload type setting.